Coupling Vanishing Point Tracking
with Inertial Navigation to
Estimate Attitude in a Structured Environment

THESIS

Dayvid Prahl, Captain, USAF

AFIT/GE/ENG/11-33

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT/GE/ENG/11-33

COUPLING VANISHING POINT TRACKING
WITH INERTIAL NAVIGATION TO
ESTIMATE ATTITUDE IN A STRUCTURED ENVIRONMENT

THESIS

Presented to the Faculty

Department of Electrical and Computer Engineering

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the

Degree of Master of Science in Electrical Engineering

Dayvid Prahl, B.S.M.E.

Captain, USAF

March 2011

COUPLING VANISHING POINT TRACKING
WITH INERTIAL NAVIGATION TO
ESTIMATE ATTITUDE IN A STRUCTURED ENVIRONMENT

Dayvid Prahl, B.S.M.E.

Captain, USAF

Approved:

_____
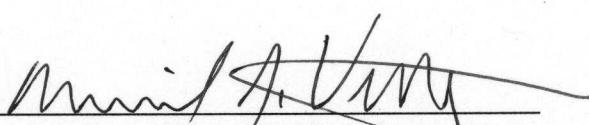
Lt Col Michael J. Veth, PhD (Chairman)

3 MAR 11

date

_____
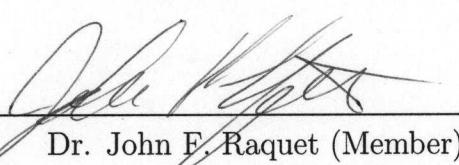
Dr. John F. Raquet (Member)

8 MAR 11

date

_____
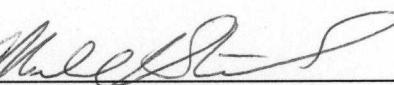
Lt Col Michael J. Stepaniak, PhD
(Member)

8 Mar 11

date

AFIT/GE/ENG/11-33

Abstract

    The Air Force's need for accurate navigation information is often met through the combination of inertial and global positioning systems. However, the increased use of smaller and smaller unmanned aerial systems opens the possibility of flight in environments where satellite navigation signals are either significantly degraded or unavailable entirely, such as indoors, in dense urban areas, and underground. Fortunately, the intrinsic orthogonal structure of man-made environments can be exploited to aid in determining a vehicle's attitude when satellite signals are unavailable. This research aims to obtain accurate and stable estimates of a vehicle's attitude by coupling consumer-grade inertial and optical sensors. This goal is pursued by first modeling both inertial and optical sensors and then developing a technique for identifying vanishing points in perspective images of a structured environment. The inertial and optical processes are then coupled to enable each one to aid the other. The vanishing point measurements are combined with the inertial data in an extended Kalman filter to produce overall attitude estimates. This technique is experimentally demonstrated in an indoor corridor setting using a motion profile designed to simulate flight. Through comparison with a tactical-grade inertial sensor, the combined consumer-grade inertial and optical data are shown to produce a stable attitude solution accurate to within 1.5 degrees. A measurement bias is manifested which degrades the accuracy by up to another 2.5 degrees.

Acknowledgements

Table of Contents

List of Figures

List of Tables

List of Symbols

List of Abbreviations

Coupling Vanishing Point Tracking with Inertial Navigation to Estimate Attitude in a Structured Environment

I. Introduction

The United States Air Force depends on precision navigation to accomplish its mission. To help fulfill this need, many of the air vehicles used by the Air Force are equipped with inertial navigation systems (INSs). While these inertial systems provide useful information regarding position and attitude in the short term, even the most advanced are subject to ever-increasing errors. Conventionally, these errors are mitigated by augmenting the inertial system with information from another source such as the global positioning system (GPS). Unfortunately, the absence of an alternate technology for constraining inertial error growth induces a dependency on the GPS. The Chief of Staff of the Air Force has addressed this dependency, stating, "It seems critical to me that the Joint force should reduce its dependence on GPS-aided precision navigation and timing, allowing it to ultimately become less vulnerable, yet equally precise, and more resilient" [23].

1.1 Unmanned Aerial Vehicles

Unmanned aerial vehicles (UAVs) are among the assortment of military assets which have come to rely on the GPS for precision navigation. Such vehicles have proven effective at providing tactical advantages in the recent conflicts on Iraq and Afghanistan and will likely continue to be a critical part of the United States arsenal for years to come. Some of these vehicles, such as the RQ-11 Raven, are small enough to be carried and hand-launched by a single soldier. As smaller and smaller UAVs are developed, they will become capable of flight in environments which have historically been unopen to aerial vehicles, such as inside buildings or underground.

1.1.1 Quadrotors. Rotary-wing vehicles have certain advantages over fixed-wing aircraft when it comes to operating in tight quarters. Specifically, slow flight and hover capabilities eliminate the need to maintain forward velocity to produce lift. One form of miniature rotary vehicle that has become more common for scientific research in recent years is the quadrotor [16], [13]. These vehicles use four counter-rotating fixed-pitch propellers to generate lift. Desired motion is obtained by simply varying the propellers' rotational speeds. Accurate attitude knowledge is requisite to controlling a quadrotor, as any rotation is immediately converted to translational motion by the vehicle's dynamics.

To facilitate navigational research, the Advanced Navigation Technology (ANT) Center at the Air Force Institute of Technology (AFIT) has developed several air vehicle platforms, including quadrotors. One such vehicle is depicted in Figure 1.1. This vehicle is equipped with a lightweight micro electro-mechanical systems (MEMS) inertial measurement unit (IMU) and a commercial webcam which can be used to augment the inertial data in the place of a GPS receiver.
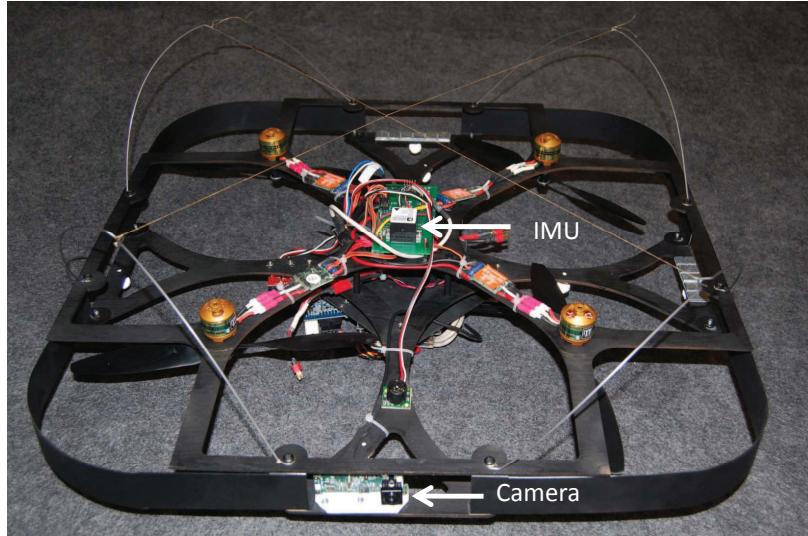
Figure 1.1: Quadrotor (bottom view). The quadrotor shown here was developed by the ANT Center as a research platform for vision-based aerial navigation solutions. It is equipped with both a MEMS inertial unit and commercial webcam.

1.2 Indoor Operation

The indoor operating environment poses distinct challenges to aerial vehicles. Tight quarters provide little room for maneuvering, and obstructions abound. To add to the difficulties, the GPS signal is substantially degraded indoors. This research aims to provide an alternative to the GPS for augmenting inertial attitude estimates in an indoor environment.

1.2.1 Manhattan World Assumption.   Many man-made environments such as dense urban or indoor settings have a consistent, orderly structure which can be exploited to obtain attitude information from an optical sensor. The attitude estimation technique described in this thesis is intended for use in such environments. The following assumptions regarding the structure of the indoor environment in which an aerial vehicle is to operate are used to enable the techniques described herein:

- All floors and ceilings are flat and level.

- All walls are flat and vertical.

- All rooms and corridors meet at right angles.

Structural features of such an environment will align to an orthogonal three-dimensional (3-D) grid described as the "Manhattan world" in [7].

1.2.2 Vanishing Points.   Most of the lines defining the edges and intersections of planar surfaces in the Manhattan world are aligned to one of three mutually orthogonal directions, forming large groups of mutually parallel lines. All of the lines in any of these groupings will appear to converge at a single point, called a vanishing point, in perspective images of a Manhattan world scene. The positions of these vanishing points in an image are shown in [11] to be invariant to translational motion of the camera, and only change when the camera is rotated.

1.3 Problem Formulation

The problem that we are trying to solve is this: attitude estimates provided by the IMU aboard our vehicle are subject to drift, which leads to an unbounded increase in attitude error. We wish to use inertial data to aid in finding vanishing points in perspective images of a Manhattan world environment and, in turn, use the positions of these vanishing points to constrain the long-term drift in the inertial attitude estimates. This will be accomplished by combining the inertial and optical data in an extended Kalman filter.

1.4 Research Contributions

The primary contribution of this research is a deeply coupled vanishing point and inertial attitude estimation method. The tight coupling of the inertial and optical sensors used for this research results in an overall attitude solution unattainable using either sensor alone. Using the methods described herein, a drift-free attitude solution accurate to within 1.5 degrees 1-σ and biased by only 2 degrees is obtained using a pair of small, inexpensive, light-weight, low-power sensors.

1.5 Outline

The remainder of this thesis is organized in the following manner. Chapter II discusses the mathematical and topical backgrounds behind inertial navigation and optical sensors, as well as other researchers' key contributions related to them. Chapter III outlines the manner in which the coupled inertial and optical attitude estimation technique presented herein was developed and the experimental procedures used to evaluate it. The experimental results are presented and analyzed in Chapter IV, and conclusions are made with a few suggestions for future work in Chapter V.

II.   Background

This chapter outlines the concepts one must understand in order to implement the attitude estimation technique described in this thesis. The notational conventions used herein are presented in Section 2.1, followed by basic concepts in terrestrial navigation in Section 2.2. Next, inertial navigation techniques and their associated limitations are described in Section 2.3. Computer vision techniques will be used to aid the navigation solution provided by an inertial sensor, so concepts in digital imaging and image processing are presented in Sections 2.4 and 2.5. Finally, methods for combining data from multiple sources are described in Section 2.6. Other researchers' contributions to the body of knowledge in these subject areas are also briefly discussed throughout.

2.1   Notation

Certain conventions are used throughout the body of this work pertaining to how variable quantities are represented. These conventions are intended to help the reader understand the quantities expressed in the equations, figures, tables, and text of this work, and are as follows:

Scalars: Scalars are represented by either upper or lowercase characters in *italics*, e.g., a or A.

Vectors: Vector quantities are represented by lowercase characters in **bold**, e.g., $\boldsymbol{a}$ or $\boldsymbol{\psi}$. Unless specifically stated otherwise, all vectors should be interpreted as column vectors.

Vector Components: The scalar components of a vector are represented with subscripts indicating their corresponding axes, e.g., the x-component of the vector $\boldsymbol{a}$ is represented as a_x.

Homogeneous Vectors: Homogeneous vectors are defined to have a final component equal to 1 and are represented with an underscore, e.g., $\underline{\boldsymbol{a}}$.

5

Skew-symmetric Matrices: The skew-symmetric matrix form of 3-vectors is sometimes useful for mathematical computations involving matrices and/or vector cross products. Vectors represented in skew-symmetric matrix form are followed by the "cross" character, e.g., $\boldsymbol{a}\times$. A vector $\boldsymbol{a}$ with components a_x, a_y, and a_z is described in skew-symmetric matrix form as shown in Equation (2.1).

$$\boldsymbol{a}\times \triangleq \begin{bmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{bmatrix} \tag{2.1}$$

Matrices: Matrices are represented by uppercase characters in **bold**, e.g., $\boldsymbol{A}$ or $\boldsymbol{\Psi}$.

Estimated Variables: Variables which represent an estimate of a particular quantity are represented with the "hat" character, e.g., $\hat{\boldsymbol{a}}$.

Corrupted Variables: Variables which are corrupted by errors are represented with the "tilde" character, e.g., $\tilde{a}$.

Relative Motion: When relative motion is specified, combined subscripts are added to a vector to describe the motion, e.g., $\boldsymbol{p}_{ab}$ represents the relative position from frame a to frame b.

A Priori and A Posteriori Estimates: Within a Kalman filter, it is necessary to distinguish between two estimates of a random variable's mean and uncertainty which are held at the same instant in time–the *a priori* estimate that is determined without incorporating new information from a measurement update and the *a posteriori* estimate which does incorporate the measurement information. In such instances, a "minus" character superscript is added to the time variable for *a priori* estimates, and a "plus" character superscript on the time variable indicates *a posteriori* estimates, e.g., $\hat{\boldsymbol{a}}(t^-)$ or $\hat{\boldsymbol{a}}(t^+)$.

2.2 Reference Frames

Navigation information is quantified using a defined reference frame. This reference frame is used to describe position relative to a point or surface and to give a mathematical realization to vector quantities such as velocity, acceleration and torque. A particular vector will have different mathematical realizations in different coordinate systems, although the vector itself is unchanged. Throughout this text, the reference frame in which a vector is expressed is denoted with a superscript. For example, the vector $\boldsymbol{y}$ expressed in the Earth-centered, Earth-fixed (ECEF) reference frame would appear as $\boldsymbol{y}^e$. Common reference frames used in navigation and computer vision include the following:

Earth-fixed inertial frame (i-frame) - The origin is fixed at the center of the Earth, with the x and y-axes on the equatorial plane and the z-axis along the Earth's axis of rotation. The i-frame does not spin with the Earth but does follow the Earth's orbit around the sun. Though it is not a true inertial frame, for the sake of terrestrial navigation it can be considered as such.

Earth-centered Earth-fixed frame (e-frame) - The origin is fixed at the center of the Earth, with the x-axis on the equatorial plane pointing to the prime meridian, z-axis parallel to the Earth's axis of rotation, and y-axis located so as to form a right-handed orthogonal triad. Unlike the i-frame, the e-frame spins along with the Earth.

Earth-fixed Navigation frame (n-frame) - This is a locally defined reference frame with its origin determined arbitrarily. The origin is typically fixed to the Earth's surface with the x-axis pointing north, y-axis pointing east and z-axis pointing down. Often, it is also called the North-East-Down (NED) frame.

Body frame (b-frame) - The origin is usually either at the center of gravity (cg) of a moving vehicle or at the center of a triad of inertial sensors. For aircraft, it is typically defined with the x-axis out the nose, y-axis out the right wing, and z-axis out the belly, as shown in Figure 2.2.
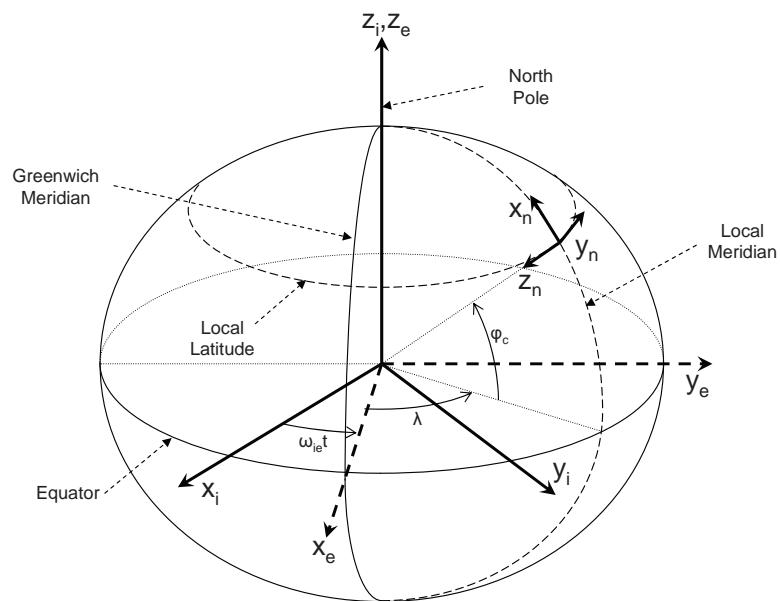
Figure 2.1: Common coordinate systems. The Earth-fixed inertial, Earth-centered Earth-fixed, and navigation frames are shown. The origins of the inertial and Earth frames are at the Earth's center of mass while the origin of the navigation frame is fixed on the Earth's surface. (Figure taken from [31])
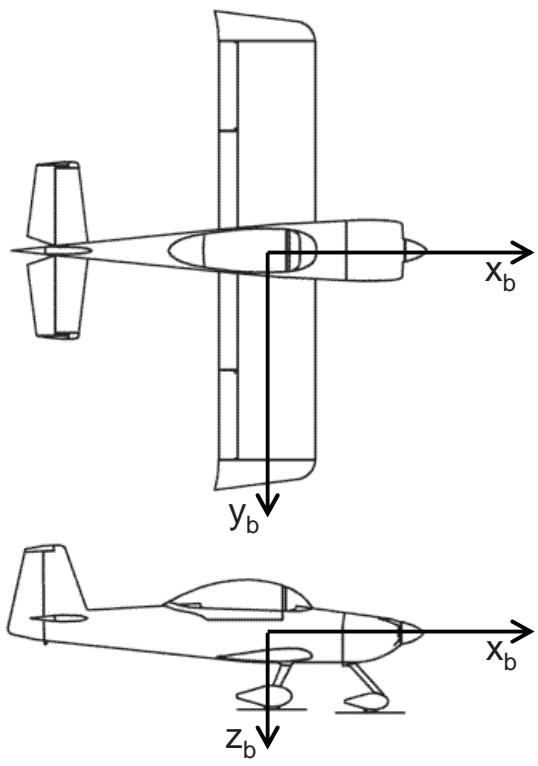
Figure 2.2:   Body reference frame. For aircraft, the b-frame is oriented with the x-axis out the nose, y-axis out the right wing and z-axis out the belly. (Figure taken from [31])

Camera frame (c-frame) - The origin is at the optical center of the camera, with
the x-axis pointed upward, y-axis to the right and z-axis out the lens, as shown
in Figure 2.3
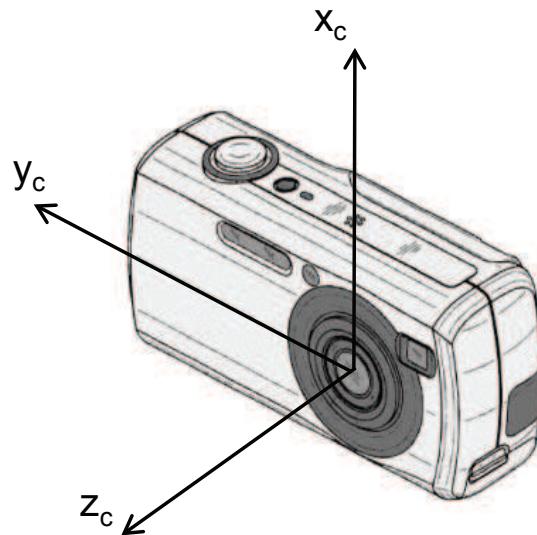
Figure 2.3:   Camera reference frame. The x-axis is upward in the images captured,
y-axis to the right, and z-axis out the lens.

Image frame (pix-frame) - Unlike the other frames mentioned, the pix-frame has
only two dimensions. The origin is beyond the upper-left pixel of a digital image,
but multiple conventions exist throughout image processing literature for pixel
indexing and axis orientation. In this work, images are indexed according to
the matrix storage format used by The Mathworks, Inc.'s Matlab software, with
the upper left pixel indexed as (1,1), the x-axis down the left side and y-axis
across the top of the image.

The navigation described in this thesis will be defined with reference to a local level
Earth-fixed navigation frame with its origin on the floor of an indoor corridor centered
between the walls, the x-axis along the length of the corridor and the z-axis pointed
down as depicted in Figure 2.4.
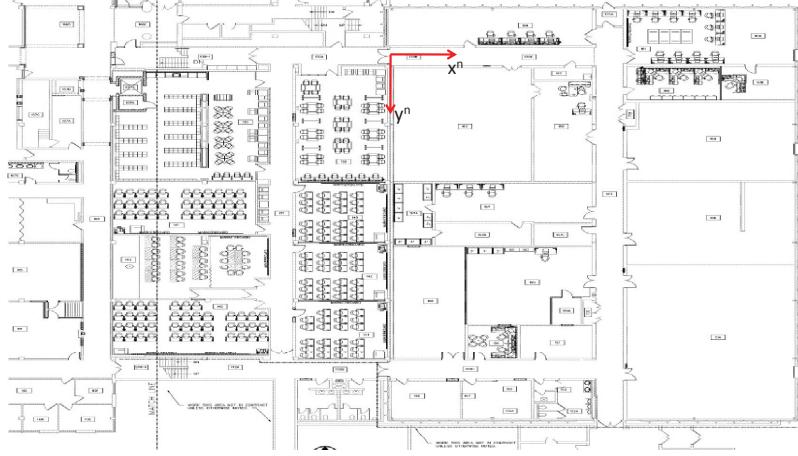
Figure 2.4: Local level navigation frame. The local level navigation frame depicted here is used as a reference for defining attitude throughout this research.

2.2.1 Coordinate Transformations. Often, it is necessary to convert vector quantities from one coordinate system to another. This is accomplished by performing a vector transformation. There are two possible ways in which right-hand orthogonal reference frames may differ from one another at a particular instant in time–translation and rotation.

2.2.1.1 Translation. If the origins of two reference frames are not collocated, a position vector is used to describe the position of one with respect to the other. In the case where two reference frames have principle axes parallel to one another but with spatially separated origins, any vector in one frame will have the same mathematical description in the other. The coordinates of a fixed point, however, will differ between the two representations. The conversion of a particular coordinate triad from one frame to the other is described by Equation (2.2), where $\boldsymbol{p}_b$ represents the position of P from the origin of frame b and $\boldsymbol{p}_a$ represents its position from the origin of frame a as depicted in Figure 2.5.

$$\boldsymbol{p}_b = \boldsymbol{p}_a - \boldsymbol{p}_{ab} \qquad (2.2)$$
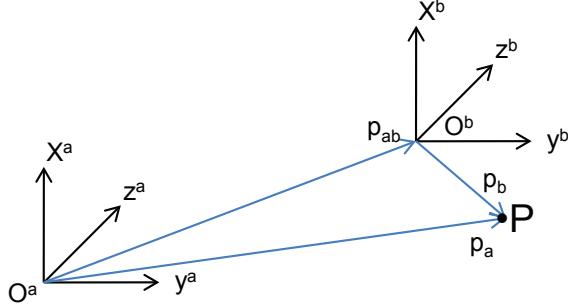
Figure 2.5:   Position vectors in two different but parallel coordinate systems. The position of point **P** from the origin of the b-frame, $\mathbf{p}_b$, is equal to the position of point **P** from the origin of the a-frame, $\mathbf{p}_a$, minus the relative position vector between frames a and b, $\mathbf{p}_{ab}$.

2.2.1.2   Rotation.   If two reference frames are oriented such that one or more of the principle axes are not co-directional, then there is a relative rotation between them.   There are several ways to represent the relative rotation between reference frames.   Two of those are direction cosine matrices (DCMs) and Euler angles.

2.2.1.2.1 Direction Cosine Matrices.   To use a direction cosine matrix to transform a vector from one frame to another, the vector is pre-multiplied by the DCM. The symbol "$\boldsymbol{C}$" is commonly used to represent a DCM, with a subscript representing the originating coordinate system and a superscript representing the destination coordinate system. As an example, the representation in a reference frame b of a vector $\boldsymbol{y}$ could be determined from its representation in frame a as shown in Equation (2.3).

$$\boldsymbol{y}^b = \boldsymbol{C}_a^b \boldsymbol{y}^a \tag{2.3}$$

A DCM's dynamics are described by the following differential equation:

$$\dot{\boldsymbol{C}}_a^b = \boldsymbol{C}_a^b [\boldsymbol{\omega}_{ba}^a \times] \tag{2.4}$$

One key property of DCMs is that the determinant is always equal to one. This ensures that a vector's magnitude is preserved when it is multiplied by the DCM.

12

Since the determinant is nonzero, this property also ensures that an inverse DCM exists. If the DCM to convert from one frame to another is known, the DCM to convert back is simply the inverse of the first, as shown in Equation (2.5). Another convenient property is that the inverse of a DCM is always its transpose, which greatly simplifies the mathematics in computing inverse DCMs.

$$C_b^a = \left[C_a^b\right]^{-1} = \left[C_a^b\right]^T \tag{2.5}$$

2.2.1.2.2 Euler Angles. Another way that relative rotation between reference frames can be expressed is through the use of Euler angles. These angles represent the following three successive rotations:

1. A rotation through angle ψ about the originating reference frame's z-axis

2. A rotation through angle θ about the new intermediate reference frame's y-axis, y'

3. A rotation through angle ϕ about the second intermediate reference frame's x-axis, x''

To obtain an equivalent DCM from a set of Euler angles, the Euler angles are each represented as a separate DCM, and the total rotation is the product of all three as shown in Equation (2.6).

$$C_a^b = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix} \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix} \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.6}$$

2.2.1.3 Transformation Matrices. Translation and rotation can be performed simultaneously through the use of homogeneous vectors and a transformation matrix. Homogeneous vectors are constructed by augmenting a vector with an additional element equal to one. For instance, the vector r with components r_x, r_y

and r_z would be converted to the homogeneous form $\underline{r}$ as shown in Equation (2.7).

$$\underline{r} = \begin{bmatrix} r_x & r_y & r_z & | & 1 \end{bmatrix}^T \tag{2.7}$$

If the rotation between a particular reference frame a and another reference frame b is described by the DCM $\boldsymbol{C}_a^b$ and the translation from b to a is described by $\boldsymbol{p}_{ba}$, then the matrix $\boldsymbol{T}_a^b$ used to transform a homogeneous vector from frame a to frame b is given by Equation (2.8).

$$\underline{r}^b \;=\; \boldsymbol{T}_a^b \underline{r}^a$$

$$\boldsymbol{T}_a^b \;=\; \begin{bmatrix} \boldsymbol{C}_a^b & | & -\boldsymbol{p}_{ab}^b \\ - & & - \\ \boldsymbol{0}_{1\times3} & | & 1 \end{bmatrix} \tag{2.8}$$

2.3 Inertial Navigation

Inertial navigation relies on the concept that starting from a known position, velocity, and attitude, a vehicle's position and attitude can be determined by sensing its motion, i.e., acceleration and rotational velocity. A typical IMU consists of at least three accelerometers that measure specific force relative to the inertial frame and three gyroscopes (or gyros) that measure either rotational acceleration or rotational velocity relative to the inertial frame.

There are two general types of IMU–either platform or strapdown. A platform INS has its sensors mounted on a platform that is gimbaled so as to always have the vertical sensor aligned with local gravity, while a strapdown IMU is rigidly mounted to the vehicle. The accelerometers and gyroscopes of a strapdown IMU are typically mounted in an orthogonal triad with their input axes parallel to the b-frame's principle axes so as to provide outputs in the body frame. Vehicles like the quadrotor for which the estimation method developed in this thesis is intended are commonly equipped

with MEMS strapdown IMUs, due the small size, weight and power requirements of such devices. Titterton and Weston thoroughly describe strapdown inertial navigation in [29].

2.3.1 *Inertial Attitude Dynamics.* Since the focus of this thesis is attitude estimation, inertial attitude calculations are described here. The quantity of interest regarding attitude is the relative rotation between the vehicle body frame and the navigation frame which can be captured by the DCM, $\boldsymbol{C}_b^n$. Applying the relationship from Equation (2.4) to this DCM gives:

$$\dot{\boldsymbol{C}}_b^n = \boldsymbol{C}_b^n [\boldsymbol{\omega}_{nb}^b \times] \tag{2.9}$$

Because the strapdown IMU provides measurements of the rotation rate between the inertial and body frames expressed in the body frame, $\boldsymbol{\omega}_{ib}^b$, an expression relating the body-to-navigation frame DCM to this rotation rate is desired. Such an expression is obtained by manipulating Equation (2.9).

The rotation rate between the n and b-frames is the difference between the inertial-to-body frame rotation rate, the rotation rate between the n and e-frames, and the Earth's sidereal rate as shown in Equation (2.10).

$$\boldsymbol{\omega}_{nb}^b = \boldsymbol{\omega}_{ib}^b - \boldsymbol{C}_n^b \boldsymbol{\omega}_{en}^n - \boldsymbol{C}_n^b \boldsymbol{C}_e^n \boldsymbol{\omega}_{ie}^e \tag{2.10}$$

Since the n-frame is fixed to the surface of the Earth, the rotation rate between the n and e-frames is always zero. This simplifies Equation (2.10) to:

$$\boldsymbol{\omega}_{nb}^b = \boldsymbol{\omega}_{ib}^b - \boldsymbol{C}_n^b \boldsymbol{C}_e^n \boldsymbol{\omega}_{ie}^e \tag{2.11}$$

Finally, converting Equation (2.11) to skew-symmetric form and substituting into Equation (2.9) yields the expression shown in Equation (2.12).

$$\dot{\boldsymbol{C}}_b^n = \boldsymbol{C}_b^n[\boldsymbol{\omega}_{ib}^b \times] - \boldsymbol{C}_e^n[\boldsymbol{\omega}_{ie}^e \times]\boldsymbol{C}_n^e \boldsymbol{C}_b^n \qquad (2.12)$$

2.3.2   Inertial Attitude Errors.   Inertial system rate gyros are subject to errors stemming from various sources. Some of these include fixed and acceleration-dependent biases, scale factor, sensor misalignment and measurement noise. All of these effects can be combined in an overall rotation model in which the measured rotation, $\boldsymbol{\omega}_{ib_m}^b$, is a function of the true rotation, $\boldsymbol{\omega}_{ib}^b$, a measurement bias, $\boldsymbol{b}^b$ and zero-mean, additive, white, Gaussian noise, $\mathbf{w}^b$.

$$\boldsymbol{\omega}_{ib_m}^b = \boldsymbol{\omega}_{ib}^b + \boldsymbol{b}^b + \mathbf{w}^b \qquad (2.13)$$

For the purposes of this research, the bias will be treated as a fixed, deterministic quantity. Though the bias might be more accurately described as a first-order Gauss-Markov process, the fixed deterministic model is justified considering the short time between image measurement updates. The strength of the noise term is determined experimentally by observing the error growth rates in multiple inertial-only attitude computations.

Due to the additive nature of the noise and bias terms in Equation (2.13), the attitude solution provided by an inertial system will drift over time. The longer the inertial system operates unaided, the larger the errors will become. This slow drift is often compensated for using an additional sensor with higher frequency error properties such as in an embedded GPS/INS (EGI). In the case of an indoor setting where GPS is unavailable, an alternative approach is required, such as the vision-aiding described in this thesis.

2.4 Digital Imaging

Digital imaging is the process by which an optical sensor converts luminous energy (light) into an array of digital data. These data can then by interpreted by a digital computer and displayed to an interested user as a photograph. In order for a digital image to be produced, light must originate from a source, reflect off the various elements of the subject, enter the aperture of an optical sensor and stimulate the sensor's photoelectric array. The output from the photoelectric array is then amplified and sampled by an analog-to-digital (A/D) converter to produce the digital image. This process is depicted in Figure 2.6.
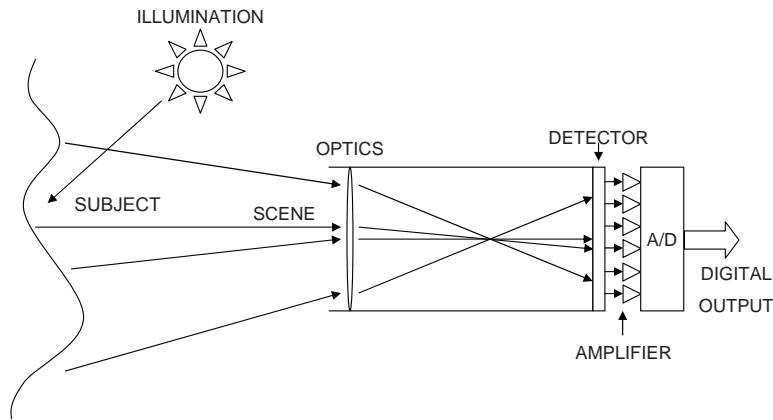
Figure 2.6:   Imaging sensor diagram. The imaging sensor interprets light reflected through the optics as a digital image. (Figure taken from [31])

2.4.1 Projective Geometry.   The process of central projection renders a 3-D subject as a two-dimensional (2-D) image. We wish to determine mathematically how to transform the coordinates of a point expressed in the 3-D camera frame into the 2-D image frame. With such a model in place, the data provided by an imaging sensor can be interpreted as it relates to the 3-D scene. This is done by modeling the effects of central projection.

2.4.1.1 Pinhole Camera.   One simple model of centralized projection is represented by a pinhole camera. In this model, all incoming light passes through

the central point, or optical center of the camera, and is projected onto a focal plane positioned at a distance of one focal length f behind the center of projection. As shown in Figure 2.6, the projection onto the focal plane is inverted by the imaging sensor's optics. The pinhole model can be further simplified to eliminate the inversion by positioning a virtual image plane one focal length in front of the center of projection. This modified pinhole camera model is shown in Figure 2.7.
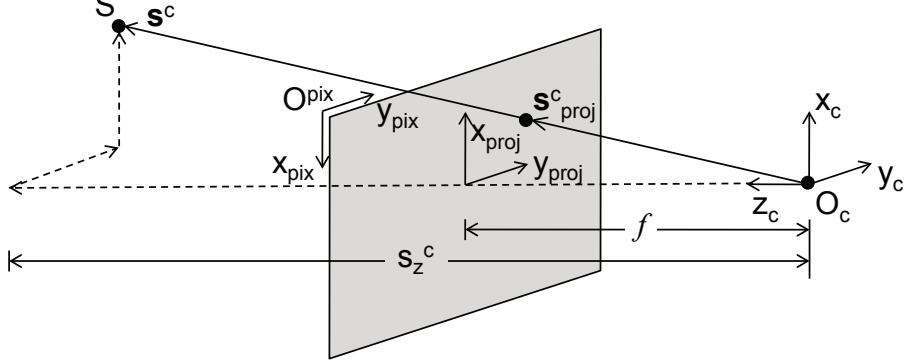
Figure 2.7: Modified pinhole camera. Light reflected from the subject crosses the virtual image plane one focal length in front of the center of projection.

A point S in the world can be identified by a position vector $\boldsymbol{s}^c$ originating at the camera center of projection and terminating at S. The vector $\boldsymbol{s}^c_{proj}$ that is co-directional with $\boldsymbol{s}^c$ but terminates at the intersection with the image plane is a scalar multiple of $\boldsymbol{s}^c$. Because the vector $\boldsymbol{s}^c_{proj}$ terminates at the image plane, its z-coordinate must be equal to f. Using the method of similar triangles, we can then determine that the scaling factor relating $\boldsymbol{s}^c$ to $\boldsymbol{s}^c_{proj}$ is f/s^c_z, as shown in Equation (2.14)

$$\boldsymbol{s}^c_{proj} = \frac{f}{s^c_z} \boldsymbol{s}^c \tag{2.14}$$

Since the image frame has only two dimensions, the transformation from camera coordinates to image coordinates must discard the z-component of the vector $\boldsymbol{s}^c_{proj}$, reducing it to a two-vector. The 2-D representation of $\boldsymbol{s}^c_{proj}$ is its projection onto the image plane, $\boldsymbol{s}^{proj}$, which originates from the projection of the camera frame's origin onto the image plane, as shown in Figure 2.8. This reduction is performed

mathematically by

$$s^{proj} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} s^c_{proj} \tag{2.15}$$

Now that s has been expressed in a 2-D frame that is coplanar with the image frame, the methods discussed in Section 2.2.1 can be applied to complete the transformation into the image frame.
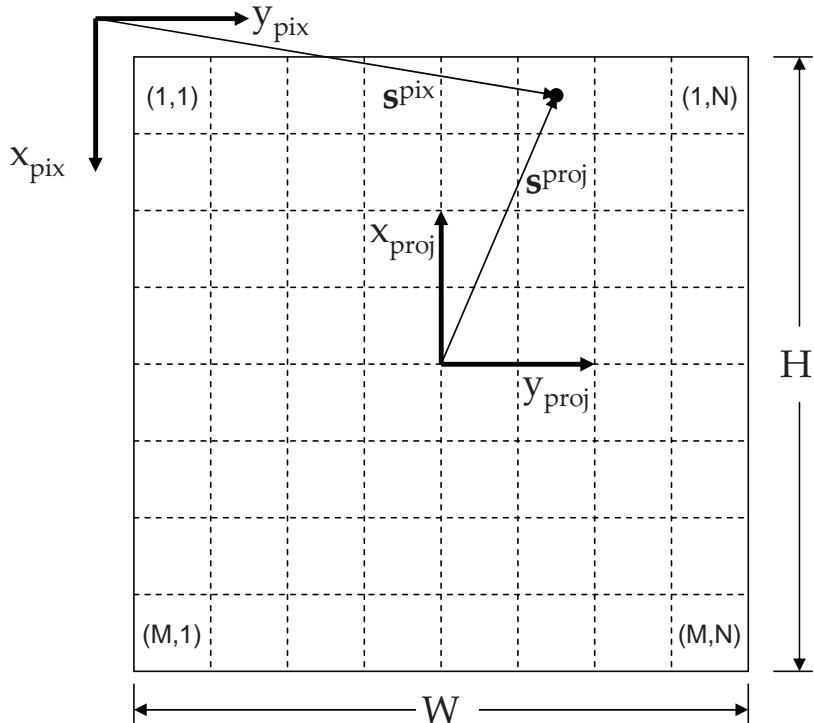
Figure 2.8:   Image plane. The image plane has physical dimensions of $H \times W$ and pixel dimensions of $M \times N$. The camera frame x and y-axes project onto the image plane centered at the coordinates $(\frac{M+1}{2}, \frac{N+1}{2})$. (Figure taken from [31])

There is a relative rotation between the image frame and projected camera frame which can be captured by a rotation matrix. Multiplying the x-coordinate by -1 will rotate a vector from the projected camera frame to the image frame as shown in Equation (2.16).

$$C^{pix}_{proj} = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \tag{2.16}$$

There is also a relative scaling factor in each direction. The focal plane has a vertical dimension of H in the camera frame and a vertical dimension of M in the image frame. This leads to a scaling factor of M/H in the x-direction. Similarly the focal plane has a horizontal dimension of W in the camera frame and a horizontal dimension of N in the image frame. This leads to a scale factor of N/W in the y-direction. These scale factors appear on the diagonal of the transformation matrix shown in Equation (2.17).

Lastly, there is a relative translation between the image frame and the projected camera frame. The origin of the camera frame projects onto the focal plane at the coordinates $(\frac{M+1}{2}, \frac{N+1}{2})$ in the image frame, which give the relative translation between the two frames. This projection of the optical center onto the focal plane is known as the principal point. The transformation of $\boldsymbol{s}^c_{proj}$ into the image frame is now given by Equation (2.17)

$$\boldsymbol{s}^{pix} = \begin{bmatrix} -\frac{M}{H} & 0 & 0 \\ 0 & \frac{N}{W} & 0 \end{bmatrix} \boldsymbol{s}^c_{proj} + \begin{bmatrix} \frac{M+1}{2} \\ \frac{N+1}{2} \end{bmatrix} \tag{2.17}$$

Substituting the right hand side of Equation (2.14) into Equation (2.17) and using homogeneous coordinates gives the total transformation from the camera frame to the image frame.

$$\underline{\boldsymbol{s}}^{pix} = \frac{1}{s^c_z} \boldsymbol{T}^{pix}_c \boldsymbol{s}^c \tag{2.18}$$

The transformation matrix, $\boldsymbol{T}^{pix}_c$, is given by:

$$\boldsymbol{T}^{pix}_c = \begin{bmatrix} -f\frac{M}{H} & 0 & \frac{M+1}{2} \\ 0 & f\frac{N}{W} & \frac{N+1}{2} \\ 0 & 0 & 1 \end{bmatrix} \tag{2.19}$$

and is known as the intrinsic camera matrix.

Since an imaging sensor is to be used to aid in determining attitude, an inverse transformation is also needed. This inverse transformation converts a pair of pixel coordinates in the image frame into a vector pointing from the center of the camera to

that pixel in the camera frame. The loss of dimension that occurs when an image is produced prevents the vector s^c from being fully determined by the inverse transformation. Instead, pre-multiplying Equation (2.18) by the inverse of the transformation matrix will only yield the homogeneous three-vector $\underline{s}^c$ that is co-directional with s^c as shown in Equation (2.20).

$$\underline{s}^c = \frac{1}{s_z^c} s^c = \boldsymbol{T}_{pix}^c \underline{s}^{pix} \tag{2.20}$$

$$\boldsymbol{T}_{pix}^c = [\boldsymbol{T}_c^{pix}]^{-1} = \begin{bmatrix} -\frac{H}{fM} & 0 & \frac{H(M+1)}{2fM} \\ 0 & \frac{W}{fN} & -\frac{W(N+1)}{2fN} \\ 0 & 0 & 1 \end{bmatrix} \tag{2.21}$$

2.4.2   Camera Calibration.   Beyond the simple pinhole camera model just described, other nonlinear distortions are present in any imaging system. These distortions give rise to the need of a calibration procedure which compensates for their effects. Such a procedure begins by modeling the distortion.

2.4.2.1   Radial Distortion.   The most visible form of distortion is radial distortion which causes the projections of straight lines to appear curved in images. This distortion occurs when the sensor's optics non-uniformly magnify the image. In [4], Brown models this distortion as a power series of r, the Euclidean length of s^{proj}, as shown in Equation (2.22), where $d^{(rad)}$ is the radial distortion factor and k_i are constant coefficients.

$$d^{(rad)} = \left(1 + k_1 r^2 + k_2 r^4 + k_3 r^6\right) \tag{2.22}$$

$$r^2 = \left(s_x^{proj}\right)^2 + \left(s_y^{proj}\right)^2 \tag{2.23}$$

2.4.2.2   Tangential Distortion.   Tangential distortion causes the principal point c to be positioned away from the geometric center of the image plane. This distortion arises from 1) imperfections in lens manufacture that cause the centers

of curvature of the front and back surfaces to not be collinear and 2) misalignment between the imaging sensor's optics and photosensitive array. Brown models this distortion as the vector function of r, s_x^{proj} and s_y^{proj} shown in Equation (2.24), where $\boldsymbol{d}^{(tan)}$ is the tangential distortion vector and p_i are constant coefficients.

$$\boldsymbol{d}^{(tan)} = \begin{pmatrix} 2p_1(s_x^{proj})(s_y^{proj}) + p_2\left[r^2 + 2(s_x^{proj})^2\right] \\ p_1\left[r^2 + 2(s_y^{proj})^2 + 2p_2(s_x^{proj})(s_y^{proj})\right] \end{pmatrix} \tag{2.24}$$

2.4.2.3   *Skew Factor.*   The skew factor, α_c, refers to the orthogonality of the pixel array's x and y-axes and accounts for the possibility that the imaging array is non-rectangular. For most cameras, the skew factor is nearly zero. The greater the skew factor, the further from 90°the angle between the image frame's x and y-axes is. When present, the skew factor appears in the upper middle position of the matrix $\boldsymbol{T}_c^{pix}$ as shown in Equation (2.25).

$$\underline{\boldsymbol{s}}^{pix} = \frac{1}{s_z^c}\begin{bmatrix} -f\frac{M}{H} & -\alpha_c f\frac{M}{H} & c_x^{pix} \\ 0 & f\frac{N}{W} & c_y^{pix} \\ 0 & 0 & 1 \end{bmatrix}\boldsymbol{s}^c \tag{2.25}$$

2.4.2.4   *Distorting Camera Model.*   Combining all of the effects discussed in the previous three sections yields a camera model that distorts the projection of $\boldsymbol{s}^c$ onto the image plane and then converts the distorted projection into pixel coordinates. This complete model is shown in Equation (2.26).

$$\underline{\boldsymbol{s}}^{pix} = \frac{1}{s_z^c}\begin{bmatrix} -f\frac{M}{H} & -\alpha_c f\frac{M}{H} & c_x^{pix} \\ 0 & f\frac{N}{W} & c_y^{pix} \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} d^{(rad)} & 0 & d_x^{(tan)} \\ 0 & d^{(rad)} & d_y^{(tan)} \\ 0 & 0 & 1 \end{bmatrix}\boldsymbol{s}^c \tag{2.26}$$

Unfortunately, an inverse map to get from distorted pixel coordinates to a camera-frame line of sight vector cannot be determined in closed form. "Because of the high degree distortion model, there exists no general algebraic expression for this in-

verse map" [3]. Instead, the distortion removal is performed using iterative numerical methods.

2.4.2.5 *Calibration.* With a distortion model defined, the camera can be calibrated to determine the distortion parameters. These parameters include the radial distortion coefficients, k_i, the tangential distortion coefficients, p_j, the pixel coordinates of the principal point, (c_x^{pix}, c_y^{pix}) and the skew factor, α_c. They are determined by photographing a subject containing features with known coordinates, such as the calibration board shown in Figure 2.9, in varying orientations and observing the difference between the actual projections of the features onto the image plane from those predicted by the pinhole camera model. The calibration is only valid for a particular focal length and zoom setting. Bouguet's Camera Calibration Toolbox for Matlab is one tool that can be used to determine the distortion parameters [3]. Besides the distortion parameters already mentioned, the toolbox also provides values for the focal length measured in both vertical and horizontal pixels (in case the pixels are not square) which appear in the upper left and center positions, respectively, of the matrix $\boldsymbol{T}_c^{pix}$.

2.5 Digital Image Processing

Now that a camera model has been developed which describes how images are produced, techniques used to process the data the images provide will be presented. Methods of edge and line detection are presented, followed by their application towards the detection of vanishing points. Lastly, methods of determining camera attitude relative to a scene based on the projections of vanishing points onto the image plane are presented.

2.5.1 *Edge Detection.* One common preprocessing step in analyzing digital images is to find edges, i.e., points where the magnitude of the gradient is high in one direction as compared with the rest of the image. Pixels where this gradient magnitude is above a particular threshold are identified as edges, or edgels.
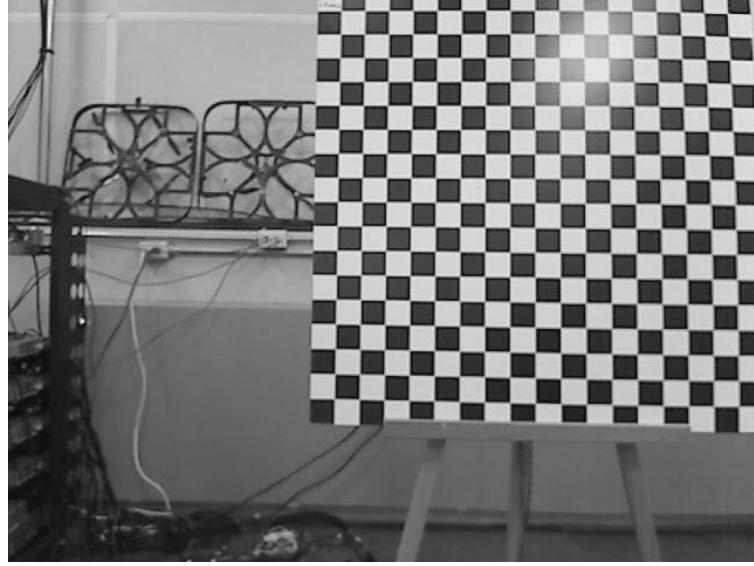
Figure 2.9: Calibration image. A calibration board such as this is used to determine a camera's distortion parameters. The effects of radial distortion are apparent in the upper right corner where the rightmost column of squares appears to curve inward.

Rather than actually computing a derivative of the image, often the gradient is approximated by evaluating the convolution of the image with a small kernel or "mask". Common convolution kernels used for this task include those proposed by Roberts [25], Prewitt [24], and Sobel [28] shown in Figure 2.10. Convolution of the digital image with the first mask of each pair produces the gradient in the vertical direction, G_V, and convolution with the second mask produces the gradient in the horizontal direction, G_H.

(a) Roberts masks   (b) Prewitt masks   (c) Sobel masks

Figure 2.10: Common gradient operators. Convolution kernels such as these are commonly used to approximate the derivative of digital images. The 2s in the center column and row of the Sobel kernels provide a smoothing effect which is helpful in suppressing noise.

24

As Gonzalez and Woods [10] observed, 2×2 masks are simple computationally, but the symmetry about a center point offered by odd-dimensioned masks is more useful for determining edge directions. Larger masks provide more accurate approximations of the derivative, since they incorporate more information into each calculation. However, larger masks also require many more computations, since for a pair of N×N masks, $2N^2$ products and $2(N^2 - 1)$ sums must be performed for each pixel.

The true gradient magnitude is determined by evaluating the Euclidean norm of the vertical and horizontal gradients for each pixel, but the less computationally expensive method of simply adding their absolute values as shown in Equation (2.27) is commonly used when constrained by data processing capacity.

$$\|\boldsymbol{G}(i,j)\| \approx |\boldsymbol{G}_V(i,j)| + |\boldsymbol{G}_H(i,j)| \qquad (2.27)$$

In [6], Canny proposed that strong and weak edges be determined by establishing both upper and lower gradient thresholds. Strong edges occur where the magnitude of the image gradient is above the upper threshold. Weak edges occur where the gradient is between the upper and lower thresholds. Only weak edges which are adjacent to strong edges are declared as edgels in the final edge image.

Regardless of the method used, the end result of an edge detection operation is a binary image, where ones represent pixels that are declared as edges and zeros represent all other pixels. Figure 2.11 shows the result of performing the Canny edge detection operation on an image of a hallway.

2.5.2 *Line Detection.* The problem of identifying straight lines in digital images has been investigated by many researchers and a plenitude of methods have been developed. However, most methods are at least loosely based on either the Hough transform from [12] or Burns' line extractor from [5].
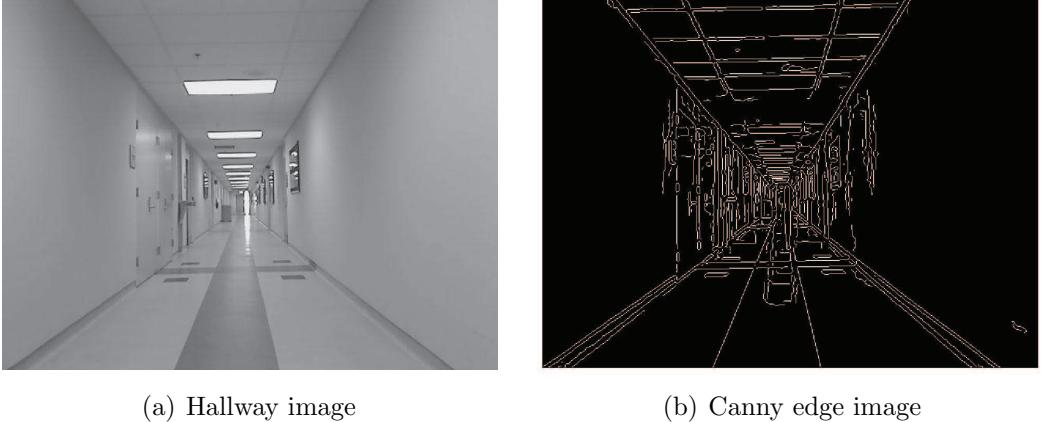
(a) Hallway image   (b) Canny edge image

Figure 2.11:   Canny edge image. (a) An image of a hallway. (b) This binary image results from performing Canny edge detection on the image shown in (a).

2.5.2.1 Hough Transform Methods.   The Hough transform was developed and patented by Paul Hough in 1962 as a means for identifying complex patterns. The method involves establishing an n-parameter representation of the pattern being sought, and then generating an n-dimensional accumulator space for determining how many observations support the presence of the pattern. The accumulator space is essentially a histogram, in which peaks appear that support likely instances of the pattern being sought.

In order to use the Hough transform to find lines in digital images, an appropriate parameterization of a line must be selected. Lines in two-dimensional image space have only two degrees of freedom, so only two parameters are required to describe them. Often, lines are represented by a slope, m, and y-intercept, b, as shown in Equation (2.28).

$$y = mx + b; \qquad\qquad\qquad (2.28)$$

However, this parameterization presents difficulties when the Hough transform is applied, because the slope parameter is unbounded, as manifest by the infinite slope of vertical lines.

Duda and Hart proposed the polar representation of a line given in Equation (2.29), which provides a fully bounded parameter space [8].

$$\rho = x\cos\theta + y\sin\theta \tag{2.29}$$

The parameters ρ and θ represent the length and angular distance from the x-axis of the shortest line segment joining the origin of a digital image to the line being observed, as shown in Figure 2.12. The parameter, θ, can vary between $\pm 90°$, and
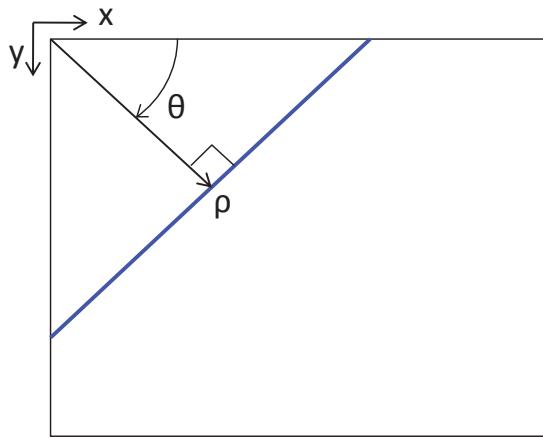
Figure 2.12:   Hough line parameters. The parameters ρ and θ represent the length and angular distance from the x-axis of the shortest line segment joining the origin of a digital image to the line being observed.

the parameter, ρ, can vary between $\pm d$, where d is the diagonal of the image frame in pixels.

An edgel with coordinates (x, y) supports the presence of every line whose parameters satisfy Equation (2.29). Thus, points in the image are represented by sinusoidal curves in the Hough parameter space, as shown in Figure 2.13. Conversely, points in Hough parameter space represent lines in the image. When many collinear edgels are present in an image, their representations in Hough parameter space stack on top of one another in the accumulator. Peaks in the accumulator correspond to pairs of parameter values representing lines in the original image, as shown in Figure 2.14.
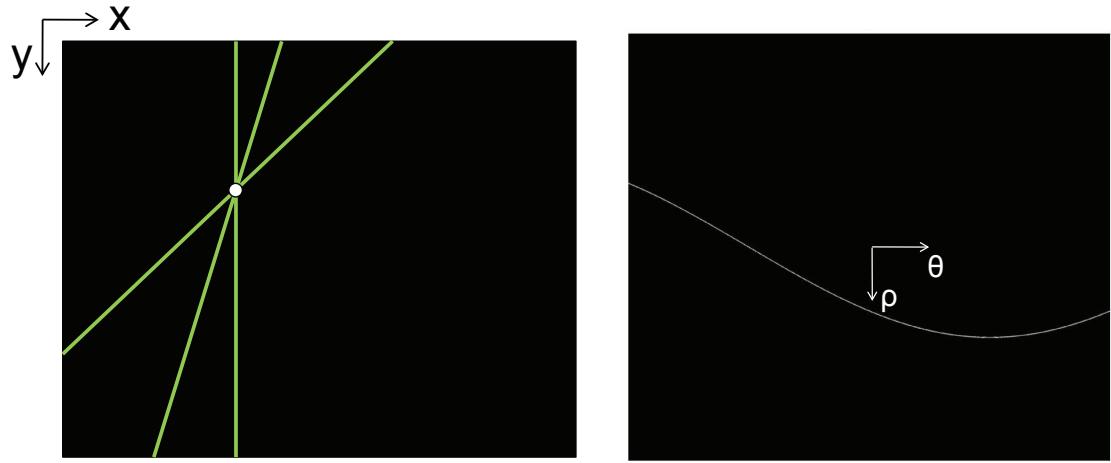
(a) Subset of lines supported by an edgel

(b) Hough parameter space for a single edgel

Figure 2.13: Hough transform of a single edgel. (a) A small subset of the lines supported by a single edgel are displayed. (b) The edgel shown in (a) is represented by the curve shown here in Hough parameter space.

(a) Collinear edgels

(b) Hough parameter space of multiple edgels
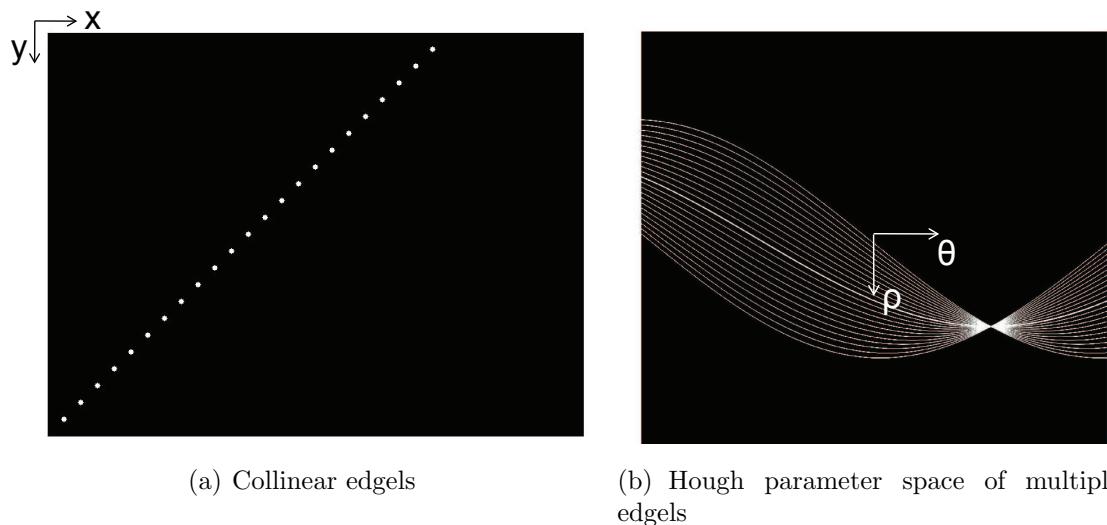
Figure 2.14: Hough transform of multiple collinear edgels. (a) All edgels shown here are collinear. (b) The edgels shown in (a) are represented by these curves in Hough parameter space. The individual curves all overlap at the point corresponding to the parameters of the line passing through the edgels in (a).

Several variations of the Hough transform have been developed, including the probabilistic Hough transform presented in [18]. Kiryati et al.'s method involves selecting a random subset of the collection of edgels in the image and performing the Hough transform on only those points. Since strong instances of a particular pattern in an image will be represented by large peaks in the Hough parameter space, the smaller subset will still identify these instances with high likelihood. This sampling method reduces the number of computations required without significantly impacting the performance in terms of pattern detection.

2.5.2.2 Gradient Orientation and Connected Component Methods.

In [5], Burns, Hanson and Riseman demonstrate that edge orientation carries important information about the presence of lines, and can be exploited for line identification. They observed that lines are characterized by neighboring pixels whose gradient orientations are roughly orthogonal to the line. The Burns line extractor uses this characteristic to fit lines to regions of neighboring pixels with similar gradient directions. All gradient orientation line detection algorithms follow the following basic steps:

1. Group pixels by common gradient direction

2. Find collections of neighboring pixels within each grouping. These are called "line support regions."

3. Fit a line to each line support region.

Burns et al.'s method begins by using convolution kernels such as those discussed in Section 2.5.1 to calculate the image gradient. The direction of the gradient at each pixel is determined by Equation (2.30) with the output of the arctangent function corrected by quadrant.

$$\theta_j = \tan^{-1} \frac{G_{V_j}}{G_{H_j}} \tag{2.30}$$

Once a gradient direction is determined for each pixel, the pixels are sorted by gradient direction into groups representing coarsely partitioned regions of the interval

$[0, 2\pi)$ as shown in Figure 2.15. The number of divisions and boundary locations are determined arbitrarily, with some researchers recommending overlapping regions to prevent fragmentation of lines whose orientations may coincide with the division boundaries. Next, a connected components algorithm is used to find groupings of
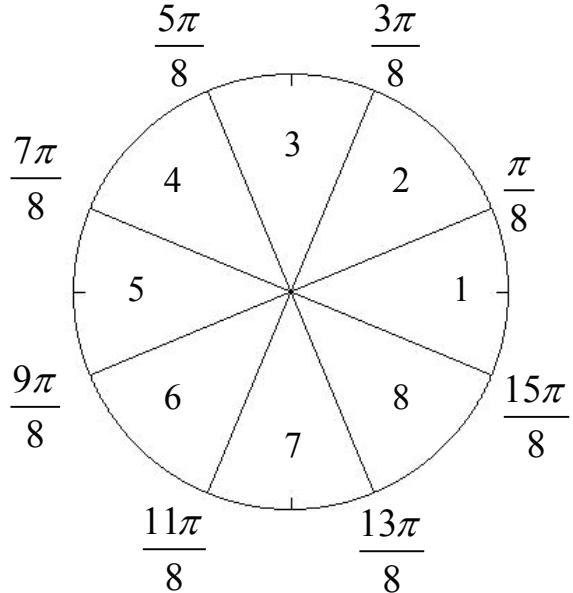
Figure 2.15: Circle divisions. Pixels are grouped by similar gradient orientation into one of the regions shown.

pixels with similar gradient orientation that are also neighbors. Each collection of connected pixels with similar gradient orientation is declared a line support region from which a straight line may be extracted.

Researchers have demonstrated different methods for fitting lines to line support regions. Burns et al. fit a plane to the intensity surface underlying each line support region [5]. They then determined lines by intersecting the intensity surface plane with a horizontal plane at the mean region intensity value. This method gives good results in terms of identifying straight lines in images and also facilitates the computation of various line characteristics such as length, contrast, width, location, orientation and straightness. Others have modified the method to enable faster image processing when computational limitations are encountered [14].

In [14], Kahn, Kitchen and Riseman developed a line extraction algorithm they call the "fast line finder" (FLF) which fits lines to line support regions by finding the major axis of an ellipse fit to the region. Their method was motivated by a need to guide a land-based robot along a path using visual cues. It is intended to be implemented on-board the mobile platform to primarily identify the edges of the path the robot is to follow.

For a line support region, R_i, composed of N individual pixels, a scatter matrix, $\boldsymbol{A}_i$, is determined by Equation (2.31)

$$\boldsymbol{A}_i = \begin{bmatrix} a_i & b_i \\ b_i & c_i \end{bmatrix} \tag{2.31}$$

where the matrix elements a_i, b_i and c_i are given by Equation (2.32).

$$
\begin{aligned}
a_i &= \sum_{j=1}^{N} w_j x_j^2 - \frac{\left(\sum_{j=1}^{N} w_j x_j\right)^2}{\sum_{j=1}^{N} w_j} \\
b_i &= \sum_{j=1}^{N} w_j x_j y_j - \frac{\left(\sum_{j=1}^{N} w_j x_j\right)\left(\sum_{j=1}^{N} w_j y_j\right)}{\sum_{j=1}^{N} w_j} \\
c_i &= \sum_{j=1}^{N} w_j y_j^2 - \frac{\left(\sum_{j=1}^{N} w_j y_j\right)^2}{\sum_{j=1}^{N} w_j}
\end{aligned}
\tag{2.32}
$$

The coordinates (x_j, y_j) give the position of each member of the line support region, and each weighting factor, w_j, is equal to the gradient magnitude of the j-th pixel. Kahn et al. opted for the standard alternative form for determining the scatter matrix shown here because it does not require the independent calculation of a region centroid prior to accumulating the partial sums.

The scatter matrix has two eigenvalues, λ_L and λ_S, with their corresponding eigenvectors, $\boldsymbol{e}_L$ and $\boldsymbol{e}_S$. One of the eigenvalues, λ_L, is most likely much larger than the other. In fact, the ratio of λ_S/λ_L can be used as a metric to describe how line-like the line support region is. The smaller the ratio, the longer and narrower the ellipse that is fit to the region.

The eigenvectors of the scatter matrix give the directions of the major and minor axes for an ellipse fit to the line support region, with $\boldsymbol{e}_L$ describing the major axis. The line, l_i, fit to line support region, R_i, is fully described by the vector passing through the centroid of the line support region whose orientation is described by Equation (2.33) with the arctangent function corrected by quadrant.

$$\theta_i = \tan^{-1}\left(\frac{\lambda_{Si} - a_i}{b_i}\right) \tag{2.33}$$

Endpoints of the line are determined by finding the intersections of l_i with the boundaries of the line support region.

In [19], Košecká and Zhang modified the line-fitting process further by omitting the weighting factors used in computing the scatter matrix. Their method more closely resembles the standard form for calculating a covariance matrix from the coordinates of the pixels comprising each line support region. First, a mean coordinate pair, $(\bar{x}_i, \bar{y}_i)$ is calculated from the coordinates of each pixel in the line support region. Then, the elements of the scatter matrix are determined using Equation (2.34).

$$
\begin{aligned}
a_i &= \sum_{j=1}^{N}(x_j - \bar{x}_i)^2 \\
b_i &= \sum_{j=1}^{N}(x_j - \bar{x}_i)(y_j - \bar{y}_i) \\
c_i &= \sum_{j=1}^{N}(y_j - \bar{y}_i)^2
\end{aligned}
\tag{2.34}
$$

Again, the eigenvalues and eigenvectors of the scatter matrix are used to determine the orientation of l_i, but Košecká and Zhang's method uses the relation shown in Equation (2.35) corrected by quadrant.

$$\theta_i = \tan^{-1}\frac{e_{Lx}}{e_{Sy}} \tag{2.35}$$

Each line is then described using the $\rho - \theta$ parameterization from Section 2.5.2.1 by using the coordinates of the centroid of each line support region, $(\bar{x}_i, \bar{y}_i)$ and the orientation, θ_i as the inputs to Equation (2.29).

2.5.2.3 Method Comparison.   In [17], Kessler et al. compare the speed of various line detection methods. Each of four different methods including the Hough transform discussed in Section 2.5.2.1 and Košecká and Zhang's connected components algorithm discussed in Section 2.5.2.2 are implemented on a 2.5 GHz computer using Matlab. Their results show that Košecká and Zhang's method of line detection is 35% faster than the Hough transform when processing a 1024×768 resolution image and 40% faster when processing a 512×384 resolution image.

2.5.2.4 Representing Image Lines in 3-Space.   In [1], Barnard describes how every line in an image can be imagined to represent a plane in 3-space which is defined by any two points on the line and the optical center of the camera. This plane, called an interpretation plane, can be described mathematically by a unit normal vector $\boldsymbol{l}_i$ as shown in Figure 2.16. The normal vector is determined by calculating the normalized cross product of the vectors pointing to the two points on the line as shown in Equation (2.36).

$$\boldsymbol{l}_i^c = \frac{\underline{\boldsymbol{s}}_1^c \times \underline{\boldsymbol{s}}_2^c}{\left\| \underline{\boldsymbol{s}}_1^c \times \underline{\boldsymbol{s}}_2^c \right\|} \tag{2.36}$$
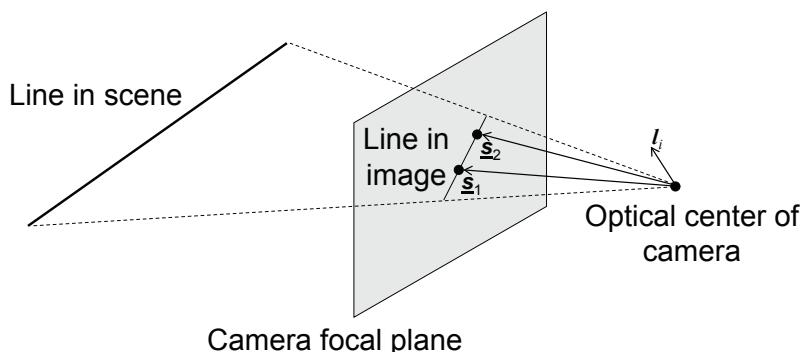
Figure 2.16:   Planar representation of an image line. Lines in images can be described by unit normals of planes in 3-space.
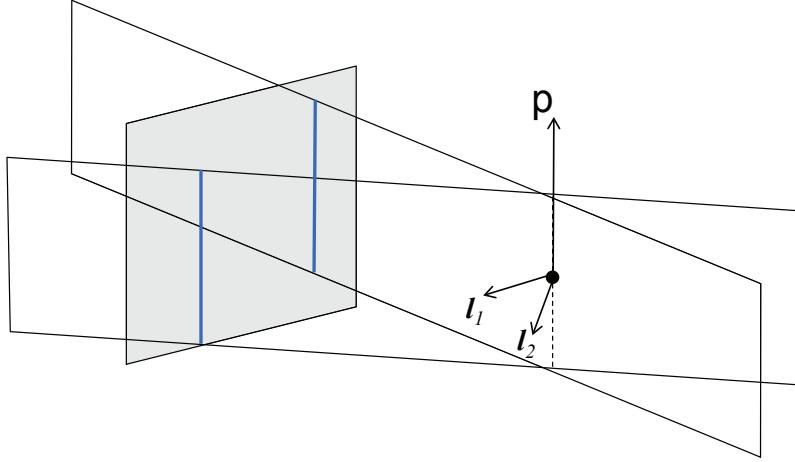
Figure 2.17: Intersection of parallel image lines. The intersection of any pair of lines in an image is expressed as the cross product of their planar normals.

This planar representation of image lines gives rise to the following three axioms:

1. The line l joining two points p_1 and p_2 is given by:

$$l = \frac{p_1 \times p_2}{\|p_1 \times p_2\|} \tag{2.37}$$

2. The point p where two lines l_1 and l_2 cross is given by:

$$p = \frac{l_1 \times l_2}{\|l_1 \times l_2\|} \tag{2.38}$$

3. Any point p on line l must satisfy:

$$p^T l = 0 \tag{2.39}$$

It is worth noting that the relationship shown in item 2 is true of any pair of lines in the image, including those that are parallel on the image plane. Such a case is illustrated in Figure 2.17. Item 3 provides a useful metric for determining how close a point is to a particular line. The greater the magnitude of their inner product, the further p is from l.

2.5.3 Vanishing Points. The projections of lines which are parallel in the world appear to converge at a fixed point in perspective images. This phenomenon is particularly visible in photographs of long corridors such as the one shown in Figure 2.11(a). The point at which parallel lines appear to intersect is known as a vanishing point, and represents the projection of a point infinitely distant from the camera onto the image plane. The projection of a vanishing point onto the image plane is invariant with relative translation, and is only affected by relative rotation between the camera and scene [11]. This property makes vanishing point detection and tracking an effective means of determining the camera's attitude with respect to a scene containing many parallel lines.

2.5.3.1 Vanishing Point Detection. A vanishing point (VP) is the point at which the projections of parallel lines in a structured scene appear to converge. Thus, the detection of vanishing points in images of structured environments is reduced to finding points where many lines intersect. To accomplish this task, Barnard proposes a method similar to the Hough transform. First an accumulator representing the Gaussian sphere, i.e., a sphere centered at the camera's optical center with radius equal to one, is constructed. The sphere is parameterized by azimuth angle, α ranging from 0 to 2π radians, and elevation angle, β, ranging from $-\frac{\pi}{2}$ to $\frac{\pi}{2}$ radians. Each element of the array represents a particular range of azimuth and elevation angles. Because these elements represent uniform angular spacing, they do not represent equal portions of the sphere's surface. Every line in the image is projected onto the Gaussian sphere by intersecting its interpretation plane with the sphere. These intersections form great circles on the sphere's surface. The accumulator is populated by incrementing the elements which satisfy Equation (2.40). (For cases where l_y is very small, an alternate form exists expressing α as a function of β with l_x in the denominator.)

$$\beta = \tan^{-1} \frac{-l_x \sin\alpha - l_z \cos\alpha}{l_y} \qquad (2.40)$$

35

Peaks in the accumulator give the azimuth and elevation of points where many lines intersect, which are likely vanishing points. Limitations of this method include the uneven spacing of accumulator elements on the sphere's surface and the ambiguous azimuth angle of vectors pointing to either pole.

In [20], Magee and Aggarwal use a similar approach, but rather than tracing circles in a discretization of the Gaussian sphere, they calculate an (α, β) pair for the intersection of each possible pairing of image lines using Equations (2.41) and (2.42).

$$\alpha = \tan^{-1}\left(\frac{p_y}{p_x}\right) \tag{2.41}$$

and

$$\beta = \tan^{-1}\left(\frac{p_z}{\sqrt{p_x^2 + p_y^2}}\right) \tag{2.42}$$

Then, groupings of intersections with an arc distance between them that is below a threshold are formed. If a large enough group is found, the associated (α, β) pair is declared a vanishing point, the lines associated with that grouping are removed from the set of image lines, and the process repeats. This method overcomes the problem stemming from the discretization of the sphere into unevenly sized segments, but still is subject to ambiguous azimuth at the poles. Both Barnard's and Magee and Aggarwal's methods also are somewhat computationally burdensome with the requirement to either trace circles on the sphere for every line or compute $_NC_2$ intersections.

2.5.3.2 *Random Sample Consensus.*   In [9], Fischler and Bolles introduce the Random Sample Consensus (RANSAC) algorithm as part of their method for determining the position of a camera based on an image of landmarks with known locations. The method provides a way to robustly fit a model to a set of data containing a certain proportion of outliers using a statistically driven guess and check scheme. When applied to the task of finding vanishing points in images, it enables the detection of clusters of mutually intersecting lines without requiring that every

possible intersection be explicitly calculated. The method is described here in general terms and applied to the detection of vanishing points in Chapter III.

Unlike conventional algorithms for fitting a model to experimental data that can be influenced by what Fischler and Boles call "gross deviations," the RANSAC algorithm is robust to such outlying data. "Rather than using as much of the data as possible to obtain an initial solution and then attempting to eliminate the invalid data points, RANSAC uses as small an initial data set as feasible and enlarges this set with consistent data when possible." [9] The process for establishing a model from a set S of datum points that is known to contain a proportion ϵ of outliers follows the basic steps outlined below.

1. Randomly select a minimum subset s from S and instantiate the model with s.

2. Determine the set of points S_i that is within a threshold t of the model established by s. The set S_i is the consensus set of S.

3. If the size of S_i is greater than a threshold T, re-estimate the model based on all the points in S_i and terminate.

4. If the size of S_i is less than T and fewer than N trials have been performed, select a new random minimum subset s and repeat steps 2 through 4.

5. After N trials, re-estimate the model with the largest consensus set S_i and terminate.

There are three unspecified parameters used in implementing the RANSAC model estimation algorithm, specifically the threshold t for declaring data as either inliers or outliers, the threshold T which determines how many data should fit the model before terminating, and the maximum number N of random minimum subsets to examine before terminating the process.

The threshold value t used for declaring data as either inliers or outliers is often determined empirically. Alternately, by assuming measurement errors are zero-mean with a known standard deviation, t can be computed from a χ^2 distribution. The size

threshold T for determining what is an adequately large consensus set is determined from the expected number of outliers as in Equation (2.43).

$$T = (1 - \epsilon)S \qquad (2.43)$$

Lastly, the minimum number of iterations N required to be assured with probability p that at least one minimum subset s is free from outliers is determined by Equation (2.44)

$$N = \frac{\log(1 - p)}{\log\left(1 - (1 - \epsilon)^s\right)} \qquad (2.44)$$

where ϵ is the proportion of outliers expected to be found in S. Naturally, p is preferred to be very nearly equal to 1, with 0.99 frequently used in practice.

In many practical applications, ϵ may not be known. Under these circumstances, the threshold T of inliers needed to end the loop cannot be determined. Instead, only the minimum number of iterations is used for deciding when to terminate the process. In order to calculate the minimum number of iterations to perform, a worst case value can be used initially, and both ϵ and N can be recomputed in subsequent iterations. If one random minimum subset s produces a proportion of outliers smaller than the current value of ϵ, N is recomputed from Equation (2.44) using the new, smaller ϵ. In the case where N is found to be smaller than the number of iterations that have already been performed, the algorithm terminates and the largest consensus set is used to estimate the model.

2.5.3.3 Determining Attitude From Vanishing Points. Various researchers have used vanishing points to determine camera orientation with respect to the scene. In [27], Schuster et al. describe the use of vanishing points to determine the heading of a ground-based robotic vehicle. Their camera is pitched upward so as to view the grid of rectangular ceiling tiles inside a building. The desired direction of travel is parallel to the short axis of the tiles, so the corresponding vanishing point is used to determine heading and make corrections in a feedback controller. Their

algorithm is limited in its scope, since only one attitude angle is determined. Also, if the image plane is parallel to the vanishing direction, a finite vanishing point cannot be found using their method.

In [13] Johnson demonstrates the use of vanishing points to find the attitude of a quadrotor unmanned vehicle within an indoor corridor. His method uses the vanishing point along the length of the corridor to directly calculate pitch θ and yaw ψ with the relationships shown in Equations (2.45) and (2.46)

$$\psi = \tan^{-1}\left(\frac{\gamma_x}{f}\right) \tag{2.45}$$

$$\theta = \tan^{-1}\left(\frac{\gamma_y}{f}\right)\cos\psi \tag{2.46}$$

where γ_x and γ_y are the distance in the camera frame's x and y directions, respectively, of the vanishing point from the principal point on the image plane. The vanishing point in the downward direction η is used to calculate the roll angle as shown in Equation (2.47).

$$\phi = -\tan^{-1}\left(\frac{\eta_y}{\eta_x}\right) \tag{2.47}$$

These attitude angles are then combined with the solution from an inertial sensor in a Kalman filter. The locations of the vanishing points in one image are used as starting points to look for vanishing points in the next image. Though a Kalman filter is used to combine the state estimates, the inertial data are not used to aid the vision routine.

In [2], Borkowski and Veth illustrate the benefits of using inertial data to predict where the vanishing point will appear in the Hough accumulator space and windowing about that point. The windowed Hough space is deemed the "predictive Hough space" and peaks corresponding to lines supporting the vanishing point are sought within it. This method is compared with inertial-only and non-predictive Hough transform methods and shown to be superior in terms of reducing attitude errors.

2.6 Kalman Filtering

With inertial and imaging methods now described, a method for merging measurements from both types of sensor is required. This combination is performed inside of a Kalman filter.

In 1960, Rudolf Kalman published his method for linear estimation in the Journal of Basic Engineering [15]. This method, which has come to be known as the Kalman filter, uses Bayesian statistics to optimally combine a dynamics model and sensor measurements to produce a minimal-uncertainty estimate of quantities of interest. An outline of Kalman's method follows, based primarily on Dr. Peter Maybeck's presentation of the topic in [21] and [22]. Though an equivalent continuous-time algorithm also exists, the Kalman filter is presented here as a discrete-time method, since it will ultimately be implemented on a digital computer.

2.6.1 Linear Kalman Filter. Some physical systems are adequately modeled in linear stochastic differential equation form as:

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{F}\boldsymbol{x}(t) + \boldsymbol{B}\boldsymbol{u}(t) + \boldsymbol{G}\mathbf{w}(t) \tag{2.48}$$

where $\boldsymbol{x}$ is a vector of state variables, $\boldsymbol{u}$ is a vector of control inputs, and $\mathbf{w}$ is a vector of zero-mean, white, Gaussian noise sources with autocorrelation:

$$E\left[\mathbf{w}(t)\mathbf{w}(t+\tau)\right] = \boldsymbol{Q}\delta(\tau) \tag{2.49}$$

The matrices $\boldsymbol{F}$, $\boldsymbol{B}$, and $\boldsymbol{G}$ are populated with constant coefficients, and $\delta(\tau)$ is the Dirac delta function. Sensor measurements for such a system may also be modeled by:

$$\boldsymbol{z}(t_i) = \boldsymbol{H}\boldsymbol{x}(t_i) + \boldsymbol{v}(t_i) \tag{2.50}$$

where z is a vector of measurements, H is a matrix of constant coefficients, and v is a vector of zero-mean, white, Gaussian noise sources with autocorrelation:

$$E\left[v(t_i)v(t_j)\right] = R\delta_{ij} \tag{2.51}$$

The symbol δ_{ij} represents Kronecker's delta function. For such a linear system, the standard Kalman filter is the optimal algorithm for estimating the random state vector, x, in terms of minimizing uncertainty in a least squares sense.

The Kalman filter provides a Gaussian probability density function (pdf) for the random vector, x, at each instant in discrete time conditioned on the uncertain measurements provided by the sensors. Beginning with initial conditions, the state estimate, $\hat{x}$, and covariance, P_{xx}, are propagated from one instant in discrete time to the next using a state transition matrix, Φ, which is determined by:

$$\Phi = e^{F\Delta t} \tag{2.52}$$

where Δt is the time step between discrete time instants. The propagation is given by:

$$\hat{x}(t_{i+1}^-) = \Phi\hat{x}(t_i^+) + Bu(t_i) \tag{2.53}$$

$$P_{xx}(t_{i+1}^-) = \Phi P_{xx}(t_i^+)\Phi^T + Q_d \tag{2.54}$$

where Q_d is the discrete-time process noise strength matrix. The calculation of Q_d is not as simple as determining Φ, but can be accomplished using the process proposed by Van Loan in [30].

At discrete instants in which measurements are available, these measurements are used to update the state estimate and covariance by optimally combining the dynamics model estimate and uncertain measurements through the use of the Kalman gain matrix, K, which is given by:

$$K(t_i) = P_{xx}(t_i^-)H^T\left[HP_{xx}(t_i^-)H^T + R\right]^{-1} \tag{2.55}$$

The dynamics model prediction and measurements are combined to produce a new state estimate and covariance using the following relationships:

$$\hat{\boldsymbol{x}}(t_i^+) = \hat{\boldsymbol{x}}(t_i^-) + \boldsymbol{K}(t_i)\left[\boldsymbol{z}(t_i) - \boldsymbol{H}\hat{\boldsymbol{x}}(t_i^-)\right] \tag{2.56}$$

$$\boldsymbol{P}_{xx}(t_i^+) = \boldsymbol{P}_{xx}(t_i^-) - \boldsymbol{K}(t_i)\boldsymbol{H}\boldsymbol{P}_{xx}(t_i^-) \tag{2.57}$$

The quantity $\boldsymbol{z}(t_i) - \boldsymbol{H}\hat{\boldsymbol{x}}(t_i^-)$ which appears in Equation (2.56) is known as the "residual," and represents the difference between the measurement realization and the measurement prediction from the dynamics model. The expression $\boldsymbol{H}\boldsymbol{P}_{xx}(t_i^-)\boldsymbol{H}^T + \boldsymbol{R}$ that appears in the Kalman gain equation (2.55) is the residual covariance. The Kalman gain serves as a weighting factor to give the appropriate amount of preference to either the dynamics model prediction or the measurement based on their respective uncertainties.

2.6.2 Extended Kalman Filtering. For systems which are not adequately modeled by Equation (2.48), the filter which has just been described will not optimally estimate the states and their uncertainties. In such a case, an extended Kalman filter (EKF) can be used to obtain more accurate estimates. The basic stochastic differential equation form of such a nonlinear system is:

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}\left[\boldsymbol{x}(t), \boldsymbol{u}(t), t\right] + \boldsymbol{G}(t)\mathbf{w}(t) \tag{2.58}$$

where $\boldsymbol{f}$ is a vector of functions, f_j, at least one of which is nonlinear. The measurement equation for such a system may also be nonlinear of the form:

$$\boldsymbol{z}(t_i) = \boldsymbol{h}\left[\boldsymbol{x}(t_i), t_i\right] + \boldsymbol{v}(t_i) \tag{2.59}$$

where $\boldsymbol{h}$ is also a vector of functions, h_k, at least one of which is nonlinear.

An alternative for overcoming the linear filter's shortfall is to linearize about the state estimates and define a vector of perturbation states. The perturbation

state vector, $\delta\boldsymbol{x}$, represents the difference between the true state and estimated state vectors.

$$\delta\boldsymbol{x}(t) \triangleq \boldsymbol{x}(t) - \hat{\boldsymbol{x}}(t) \tag{2.60}$$

The EKF produces a nominal state trajectory for each measurement interval by starting from the most recent state estimate and integrating the nonlinear state functions as shown in Equation (2.61).

$$\hat{\boldsymbol{x}}(t_{i+1}^-) = \int_{t_i}^{t_{i+1}} \boldsymbol{f}\left[\hat{\boldsymbol{x}}(t), \boldsymbol{u}(t), t\right] dt + \hat{\boldsymbol{x}}(t_i^+) \tag{2.61}$$

The estimated perturbation state vector, $\delta\hat{\boldsymbol{x}}$, is set to zero at the beginning of a filter cycle, updated during the measurement update phase, added to the nominal trajectory to correct the total state estimate, and reset to zero before beginning the next recursion.

The state transition matrix used to propagate the covariance is found by first linearizing the state functions about the most recent state estimate to obtain the matrix $\boldsymbol{F}$ as shown in Equation (2.62).

$$\boldsymbol{F}(t_i) = \left.\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}\right|_{\hat{\boldsymbol{x}}(t_i^+)} \tag{2.62}$$

This linearized state matrix is then input into Equation (2.52) to obtain $\boldsymbol{\Phi}(t_i)$, and the uncertainty is propagated with Equation (2.54).

For the measurement update stage of the filter, first a measurement prediction, $\boldsymbol{z}_{pred}$, is determined by evaluating the nonlinear measurement function at the most recent state estimate.

$$\boldsymbol{z}_{pred}(t_i) = \boldsymbol{h}\left[\hat{\boldsymbol{x}}(t_i^-), t_i\right] \tag{2.63}$$

A measurement perturbation, $\delta\boldsymbol{z}$, is also defined, which represents the difference between the actual measurements and measurement prediction as shown in Equa-

tion (2.64).

$$\delta \boldsymbol{z}(t_i) \triangleq \boldsymbol{z}(t_i) - \boldsymbol{z}_{pred}(t_i) \tag{2.64}$$

Since $\boldsymbol{z}_{pred}$ was obtained using the nonlinear measurement equations, $\delta \boldsymbol{z}$ is equivalent to the residual in the linear Kalman filter.

In order to calculate a Kalman gain matrix and residual covariance, a linearized matrix, $\boldsymbol{H}(t_i)$, must be found in much the same way that $\boldsymbol{F}(t_i)$ was. The matrix $\boldsymbol{H}(t_i)$ is obtained by evaluating the derivative of the measurement equations with respect to the state vector at the most recent state estimate as shown in Equation (2.65).

$$\boldsymbol{H}(t_i) = \left.\frac{\partial \boldsymbol{h}}{\partial \boldsymbol{x}}\right|_{\hat{\boldsymbol{x}}(t_i^-)} \tag{2.65}$$

The linearized matrix $\boldsymbol{H}(t_i)$ is then used in Equation (2.55) to calculate the Kalman gain. With the Kalman gain matrix determined, Equation (2.56) can be used to update the estimated perturbation state vector, $\delta\hat{\boldsymbol{x}}$. Since the *a priori* estimate of the perturbation state vector is zero, and the measurement perturbation is the residual, this step reduces to the form shown in Equation (2.66).

$$\delta\hat{\boldsymbol{x}}(t_{i+1}^+) = \boldsymbol{K}(t_{i+1})\delta \boldsymbol{z}(t_{i+1}) \tag{2.66}$$

Finally, the perturbation state estimate is added to the nominal state trajectory to update the total state estimate as shown in Equation (2.67), and then reset to zero for the next iteration.

$$\hat{\boldsymbol{x}}(t_{i+1}^+) = \hat{\boldsymbol{x}}(t_{i+1}^-) + \delta\hat{\boldsymbol{x}}(t_{i+1}^+) \tag{2.67}$$

Often, a certain level of pseudonoise must be added to the system to account for additional uncertainty which results from the linearization process. This is part of tuning the filter, and is typically combined with simulation to verify filter performance. Also, EKFs are somewhat sensitive to initialization errors which can prevent

effective state estimation, particularly if there is limited observability between the measurements and states.

2.7   *Summary*

This chapter has put in place the foundation upon which the attitude estimation methods used in this thesis are built. Topics in nomenclature, inertial navigation, digital imaging, and Kalman filtering have been presented. The next chapter describes how these concepts are used to develop a method for combining inertial and optical information to estimate attitude.

III. Methodology

This chapter describes the methods and procedures used to approach a solution to the indoor aerial attitude estimation problem, as well as the experimental approach used to evaluate this solution. These topics are divided into two main sections–algorithm development and experimental methods. Algorithm development is presented in Section 3.1, and experimental methods are discussed in Section 3.2.

3.1 Algorithm Development

The method presented in this thesis for estimating attitude using inertial and optical data follows the basic steps shown on the flowchart in Figure 3.1. These steps
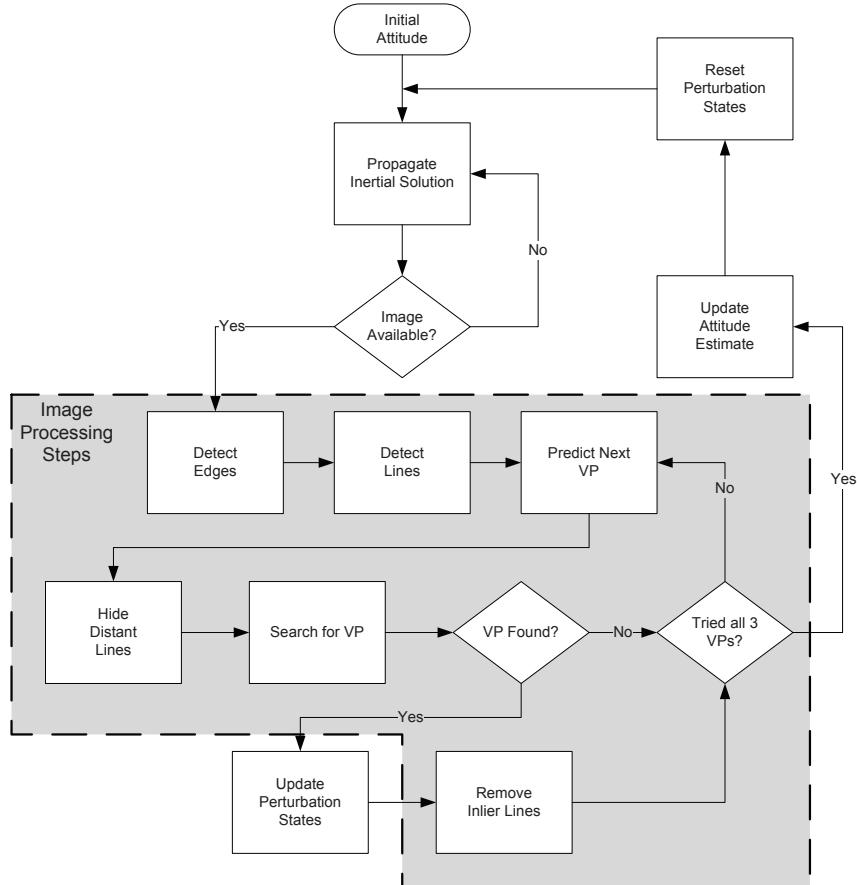
Figure 3.1:   Image-aiding algorithm flow chart. The process of combining inertial and image data follows the basic steps shown here. The steps inside the shaded region comprise the measurement stage of an EKF.

parallel the general outline of an EKF presented in Section 2.6.2, and each will be thoroughly described in the subsequent portions of this section. In order to implement such a Kalman filter, a state vector must be defined and both a dynamics model and a measurement model must be established.

3.1.1 State Vector. Attitude is expressed as the DCM which transforms vectors represented in the vehicle body frame into their equivalent representation in the navigation frame, $\mathbf{C}_b^n$. Essentially, the true body-to-navigation frame DCM is the product of a corrupted estimate of the DCM, $\mathbf{C}_b^{\tilde{n}}$, and another DCM of errors, $\mathbf{C}_{\tilde{n}}^n$, as shown in Equation (3.1).

$$\boldsymbol{C}_b^n = \boldsymbol{C}_{\tilde{n}}^n \boldsymbol{C}_b^{\tilde{n}} \tag{3.1}$$

The quantity we wish to estimate is the DCM of errors. Assuming the errors are small, they can be expressed as a 3-vector, $\boldsymbol{\psi}$, whose elements represent the rotation angles required to correct the corrupted body-to-navigation frame DCM, $\boldsymbol{C}_b^{\tilde{n}}$, to truth. This vector is the vector of perturbation states estimated by the extended Kalman filter. Its corresponding DCM can be expressed as the matrix exponential of the skew-symmetric representation of $\boldsymbol{\psi}$.

$$\boldsymbol{C}_{\tilde{n}}^n = e^{\boldsymbol{\psi}\times} \tag{3.2}$$

Since the angles contained in $\boldsymbol{\psi}$ are assumed to be small, the matrix exponential can be approximated as a matrix power series with higher than first order terms neglected. Substituting this approximation into Equation (3.1) yields the following relationship:

$$\boldsymbol{C}_b^n \approx [\boldsymbol{I} + \boldsymbol{\psi}\times]\boldsymbol{C}_b^{\tilde{n}} \tag{3.3}$$

3.1.2 System Dynamics Model. With the state vector defined, the system dynamics model can be established. This model is taken from the differential equation

governing the dynamics of $\boldsymbol{C}_b^n$ shown in Equation (2.12), which is repeated here.

$$\dot{\boldsymbol{C}}_b^n = \boldsymbol{C}_b^n [\boldsymbol{\omega}_{ib}^b \times] - \boldsymbol{C}_e^n [\boldsymbol{\omega}_{ie}^e \times] \boldsymbol{C}_n^e \boldsymbol{C}_b^n \tag{3.4}$$

Neglecting the Earth's turn rate and solving the differential equation over one discrete time-step gives the relationship shown in Equation (3.5), where Δt represents the step size between instants in discrete time.

$$\boldsymbol{C}_b^n(t_{i+1}) \approx \boldsymbol{C}_b^n(t_i) \exp\left([\boldsymbol{\omega}_{ib}^b(t_i)\Delta t] \times\right) \tag{3.5}$$

The Earth's turn rate is omitted because the update rate for this particular implementation is approximately three hertz and the error introduced by the Earth's rotation over a period of one third of a second is negligible relative to the precision of a MEMS-grade IMU. Substituting the measured body rotation rate, $\boldsymbol{\omega}_{ib_m}^b$, for $\boldsymbol{\omega}_{ib}^b$ and the estimated body-to-navigation frame DCM, $\boldsymbol{C}_b^{\tilde{n}}$, for $\boldsymbol{C}_b^n$ gives the expression shown in Equation (3.6).

$$\boldsymbol{C}_b^{\tilde{n}}(t_{i+1}^-) = \boldsymbol{C}_b^{\tilde{n}}(t_i^+) \exp\left([\boldsymbol{\omega}_{ib_m}^b(t_i)\Delta t] \times\right) \tag{3.6}$$

Equation (2.13) shows that the measured body rotation rate is the sum of the true body rotation rate, a vector of gyro biases, and zero-mean, Gaussian noise. If the vector of biases is subtracted from the measured rotation rates, a more accurate inertial solution can be obtained than the solution which results if this subtraction is not performed. Equation (3.7) shows the results of substituting the difference between the measured body rotation rate and gyro bias vector into Equation (3.6).

$$\boldsymbol{C}_b^{\tilde{n}}(t_{i+1}^-) = \boldsymbol{C}_b^{\tilde{n}}(t_i^+) \exp\left[\left([\boldsymbol{\omega}_{ib_m}^b(t_i) - \boldsymbol{b}^b]\Delta t\right) \times\right] \tag{3.7}$$

This is the nonlinear equation used to propagate $\boldsymbol{C}_b^{\tilde{n}}$ from one instant in discrete time to the next. The propagation of the perturbation state uncertainty matrix, $\boldsymbol{P}_{\psi\psi}$, will be described next.

3.1.2.1 System Model Linearization. Before the process uncertainty can be propagated, a linearized dynamics matrix must be found from which a state transition matrix can be determined. The first step in finding this linearized dynamics matrix is to differentiate Equation (3.3) with respect to time.

$$\dot{\boldsymbol{C}}_b^n = [\dot{\boldsymbol{\psi}}\times]\boldsymbol{C}_b^{\tilde{n}} + [\boldsymbol{I} + \boldsymbol{\psi}\times]\dot{\boldsymbol{C}}_b^{\tilde{n}} \tag{3.8}$$

From here, the DCM derivatives, $\dot{\boldsymbol{C}}_b^n$ and $\dot{\boldsymbol{C}}_b^{\tilde{n}}$, are replaced with their corresponding equivalent products as described by Equation (2.4).

$$\boldsymbol{C}_b^n[\boldsymbol{\omega}_{nb}^b\times] = [\dot{\boldsymbol{\psi}}\times]\boldsymbol{C}_b^{\tilde{n}} + [\boldsymbol{I} + \boldsymbol{\psi}\times]\boldsymbol{C}_b^{\tilde{n}}[\boldsymbol{\omega}_{\tilde{n}b}^b\times] \tag{3.9}$$

Substituting Equation (3.3) for $\boldsymbol{C}_b^n$ and solving for $[\dot{\boldsymbol{\psi}}\times]$ yields:

$$[\dot{\boldsymbol{\psi}}\times] = -[\boldsymbol{I} + \boldsymbol{\psi}\times]\boldsymbol{C}_b^{\tilde{n}}[\boldsymbol{\omega}_{\tilde{n}b}^b\times]\boldsymbol{C}_{\tilde{n}}^b + [\boldsymbol{I} + \boldsymbol{\psi}\times]\boldsymbol{C}_b^{\tilde{n}}[\boldsymbol{\omega}_{nb}^b\times]\boldsymbol{C}_{\tilde{n}}^b \tag{3.10}$$

Now, a replacement expression for $\boldsymbol{\omega}_{nb}^b$ will be described, starting with the following relationship:

$$\boldsymbol{\omega}_{nb}^b = \boldsymbol{\omega}_{ib_m}^b - \boldsymbol{C}_n^b\boldsymbol{C}_e^n\boldsymbol{\omega}_{ie}^e \tag{3.11}$$

Substituting Equations (2.13) and (3.3) into (3.11) gives:

$$\boldsymbol{\omega}_{nb}^b = \boldsymbol{\omega}_{ib}^b + \boldsymbol{b}^b + \mathbf{w}^b - \boldsymbol{C}_{\tilde{n}}^b[\boldsymbol{I} - \boldsymbol{\psi}\times]\boldsymbol{C}_e^n\boldsymbol{\omega}_{ie}^e \tag{3.12}$$

Distributing the product in the last term and rearranging yields:

$$\boldsymbol{\omega}_{nb}^b = \boldsymbol{\omega}_{ib}^b - \boldsymbol{C}_{\tilde{n}}^b\boldsymbol{C}_e^n\boldsymbol{\omega}_{ie}^e + \boldsymbol{b}^b + \mathbf{w}^b + \boldsymbol{C}_{\tilde{n}}^b[\boldsymbol{\psi}\times]\boldsymbol{C}_e^n\boldsymbol{\omega}_{ie}^e \tag{3.13}$$

Combining the first two terms on the right hand side of the equation gives:

$$\boldsymbol{\omega}_{nb}^{b} = \boldsymbol{\omega}_{\tilde{n}b}^{b} + \boldsymbol{b}^{b} + \mathbf{w}^{b} + \boldsymbol{C}_{\tilde{n}}^{b}[\boldsymbol{\psi}\times]\boldsymbol{C}_{e}^{n}\boldsymbol{\omega}_{ie}^{e} \tag{3.14}$$

The expression found in Equation (3.14) can now be substituted into Equation (3.10) to obtain:

$$\begin{aligned}
[\dot{\boldsymbol{\psi}}\times] &= -[\boldsymbol{I} + \boldsymbol{\psi}\times]\boldsymbol{C}_{b}^{\tilde{n}}[\boldsymbol{\omega}_{\tilde{n}b}^{b}\times]\boldsymbol{C}_{\tilde{n}}^{b} + \\
&\quad [\boldsymbol{I} + \boldsymbol{\psi}\times]\boldsymbol{C}_{b}^{\tilde{n}}\Big([\boldsymbol{\omega}_{\tilde{n}b}^{b}\times] + [\boldsymbol{b}^{b}\times] + [\mathbf{w}^{b}\times] + \big[(\boldsymbol{C}_{\tilde{n}}^{b}[\boldsymbol{\psi}\times]\boldsymbol{C}_{e}^{n}\boldsymbol{\omega}_{ie}^{e})\times\big]\Big)\boldsymbol{C}_{\tilde{n}}^{b}
\end{aligned} \tag{3.15}$$

Eliminating like terms from this expression gives:

$$[\dot{\boldsymbol{\psi}}\times] = [\boldsymbol{I} + \boldsymbol{\psi}\times]\boldsymbol{C}_{b}^{\tilde{n}}\Big([\boldsymbol{b}^{b}\times] + [\mathbf{w}^{b}\times] + \big[(\boldsymbol{C}_{\tilde{n}}^{b}[\boldsymbol{\psi}\times]\boldsymbol{C}_{e}^{n}\boldsymbol{\omega}_{ie}^{e})\times\big]\Big)\boldsymbol{C}_{\tilde{n}}^{b} \tag{3.16}$$

Finally, removing second-order terms, neglecting the Earth's turn rate and collapsing the skew-symmetric forms yields:

$$\dot{\boldsymbol{\psi}} = \boldsymbol{C}_{b}^{\tilde{n}}\boldsymbol{b}^{b} + \boldsymbol{C}_{b}^{\tilde{n}}\mathbf{w}^{b} \tag{3.17}$$

This equation shows that the perturbation angles' rates of change are simply the sum of a vector of biases and white, Gaussian noise. This, then, gives rise to a linearized state matrix that is populated with all zeros and a state transition matrix equal to identity. The error matrix, $\boldsymbol{P}_{\psi\psi}$, is then propagated from one instant in discrete time to the next by:

$$\boldsymbol{P}_{\psi\psi}(t_{i+1}^{-}) = \boldsymbol{P}_{\psi\psi}(t_{i}^{+}) + \boldsymbol{Q}_{d} \tag{3.18}$$

Together, Equations (3.6) and (3.18) comprise the "Propagate Inertial Solution" step depicted on the flowchart shown in Figure 3.1.

3.1.3 Measurement Model. Along with a system dynamics model, an EKF also requires a measurement model. In this case, the measurement model begins

with the assumptions regarding the structure of the environment enumerated in Section 1.2.1. Such an environment will have three mutually orthogonal primary vanishing directions that coincide with the three principal axes of the Earth-fixed local level navigation frame. Mathematically, this means that there will be three vanishing points, $\mathbf{v}_1$, $\mathbf{v}_2$ and $\mathbf{v}_3$, which can be expressed as the following unit pointing vectors:

$$\mathbf{v}_1^n = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \quad \mathbf{v}_2^n = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \quad \mathbf{v}_3^n = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \tag{3.19}$$

Because there are three vanishing points inherent in the structure of the environment, there are three measurements for each image. Inside the Kalman filter, these are treated as consecutive updates to the vector of perturbation states where the *a posteriori* estimate of $\hat{\boldsymbol{\psi}}$ determined from one vanishing point becomes the *a priori* estimate for the next. Once the last update to the perturbation state vector has been performed, $\boldsymbol{C}_b^{\tilde{n}}$ is corrected using the following relationship:

$$\boldsymbol{C}_b^{\tilde{n}}(t_i^+) = \exp\left([\hat{\boldsymbol{\psi}}(t_i^+)\times]\right)\boldsymbol{C}_b^{\tilde{n}}(t_i^-), \quad \hat{\boldsymbol{\psi}}(t_i^+) \to \mathbf{0}_{3\times1} \tag{3.20}$$

This corresponds to the "Update Attitude Estimate" and "Reset Perturbation States" steps shown on the flowchart in Figure 3.1. The DCM, $\boldsymbol{C}_b^{\tilde{n}}$, is not updated between the three vanishing point measurements in a single image, so that each of the three measurement predictions utilizes the same *a priori* information and all three measurements remain mutually independent.

The measurement function describing the vanishing points' representations in the camera frame is shown in Equation (3.21),

$$\mathbf{v}_k^c = \boldsymbol{C}_b^c \boldsymbol{C}_n^b \mathbf{v}_k^n + \boldsymbol{v}_k \quad \forall\ k \in [1,3] \tag{3.21}$$

51

where $\boldsymbol{v}_k$ represents the measurement noise vector corresponding to the k-th vanishing point. Substituting Equation (3.3) into Equation (3.21) yields the following expression relating the observed vanishing point vector in the camera frame to the vector of perturbation states:

$$\mathbf{v}_k^c = \boldsymbol{C}_b^c \boldsymbol{C}_{\tilde{n}}^b [\boldsymbol{I} - \boldsymbol{\psi} \times] \mathbf{v}_k^n + \boldsymbol{v}_k \quad \forall\ k \in [1,3] \tag{3.22}$$

This is the nonlinear measurement function which must be linearized in order to compute a Kalman gain for use in the update step of the Kalman filter.

3.1.3.1 Measurement Model Linearization. Before the nonlinear measurement function is differentiated, it is manipulated algebraically into a more convenient form. Distributing the matrix product on the right hand side of Equation (3.22) yields:

$$\mathbf{v}_k^c = \boldsymbol{C}_b^c \boldsymbol{C}_{\tilde{n}}^b \mathbf{v}_k^n - \boldsymbol{C}_b^c \boldsymbol{C}_{\tilde{n}}^b [\boldsymbol{\psi} \times] \mathbf{v}_k^n + \boldsymbol{v}_k \quad \forall\ k \in [1,3] \tag{3.23}$$

The product, $[\boldsymbol{\psi} \times] \mathbf{v}_k^n$, appearing on the right of Equation (3.23) represents a vector cross product. The order of multiplication can be reversed, so long as the product is multiplied by a factor of -1 as shown in Equation (3.24).

$$\mathbf{v}_k^c = \boldsymbol{C}_b^c \boldsymbol{C}_{\tilde{n}}^b \mathbf{v}_k^n + \boldsymbol{C}_b^c \boldsymbol{C}_{\tilde{n}}^b [\mathbf{v}_k^n \times] \boldsymbol{\psi} + \boldsymbol{v}_k \quad \forall\ k \in [1,3] \tag{3.24}$$

Differentiating Equation (3.24) with respect to $\boldsymbol{\psi}$ yields an expression for the linearized measurement matrices, $\boldsymbol{H}_k$.

$$\boldsymbol{H}_k(t_i) = \boldsymbol{C}_b^c \boldsymbol{C}_{\tilde{n}}^b (t_i^-) [\mathbf{v}_k^n \times] \quad \forall\ k \in [1,3] \tag{3.25}$$

3.1.3.2 Measurement Noise Strength. In order to obtain the Kalman gain matrices that are used to incorporate the measurements into the estimated state vector, the strengths of the measurement noise vectors must also be known. For this

work, the measurement noises are characterized as zero-mean, white and Gaussian with autocorrelations given by:

$$E[\boldsymbol{v}_k(t_i)\boldsymbol{v}_k^T(t_j)] = \boldsymbol{R}_k\delta_{ij} \quad \forall \ k \in [1,3] \tag{3.26}$$

where

$$\boldsymbol{R}_k = \begin{bmatrix} 0.01 & 0 & 0 \\ 0 & 0.01 & 0 \\ 0 & 0 & 0.01 \end{bmatrix} \quad \forall \ k \in [1,3] \tag{3.27}$$

This equates to a standard deviation of 0.1 in each of the measured unit pointing vectors' Cartesian coordinates.

With the linearized measurement matrices, $\boldsymbol{H}_k(t_i)$, and measurement noise strength matrices, $\boldsymbol{R}_k$, defined, Kalman gains for each measurement can be computed using Equation (2.55). These Kalman gains are then used in Equation (2.66) to update perturbation state estimate for each of the three orthogonal vanishing directions.

3.1.4 Obtaining Vanishing Point Measurements.   Now that a dynamics model and a measurement model have been established and the method for incorporating measurements into the attitude estimate is in place, all we lack are the actual measurements themselves. This section describes how measurements of each vanishing point are obtained from a digital image.

3.1.4.1 Edge Detection Method Selection.   The first step in processing an image to find vanishing points is to extract edges. As discussed in Section 2.5.1, there are various methods for performing this step. Each of the four edge detection methods presented in Section 2.5.1 were performed on several hallway images to determine which would be the preferable method to use. An image of a hallway taken with the host vehicle's camera and the results of performing all four methods of edge

detection are shown in Figure 3.2. Because the canny edge detector returned more edges, in particular useful edges such as the borders of the ceiling tiles, the canny edge detector was chosen for processing the images collected during experiments.

3.1.4.2 Line Detection Method Selection. After edges have been found in an image, the next step towards identifying the vanishing points is to extract straight lines. To this end, two of the line detection methods discussed in Section 2.5.2 were evaluated to determine which is better suited for this particular application.

While the Hough transform is a widely-used method for extracting straight lines in images, the comparison presented by Kessler, et al. in [17] shows moderately reduced computation time in using a connected-components method to accomplish this task. Because the microcomputer on board the host vehicle has limited capacity for image processing, a fast line finding technique can provide distinct advantages over other, slower methods. In light of this information, a connected components line detection process as described by Košecká and Zhang [19] was implemented and compared with the Hough transform method to determine which is faster. Both methods were executed using Matlab R2009b software on a 2.5 GHz Windows® computer. The resolution of the Hough accumulator was set at one pixel for ρ and one degree for θ, and sixteen equally-spaced gradient direction regions were used for the connected components algorithm. The running times of each line detection method for four different images captured from the quadrotor's onboard camera are shown in Table 3.1.

Table 3.1: Line detection times. Two different methods' processing times for detecting lines in several images captured with the host vehicle's camera are shown here.

Image	Connected Components	Hough Transform
Image 1	0.26s	0.37s
Image 2	0.27s	0.26s
Image 3	0.26s	0.29s
Image 4	0.28s	0.11s

(a) Hallway image
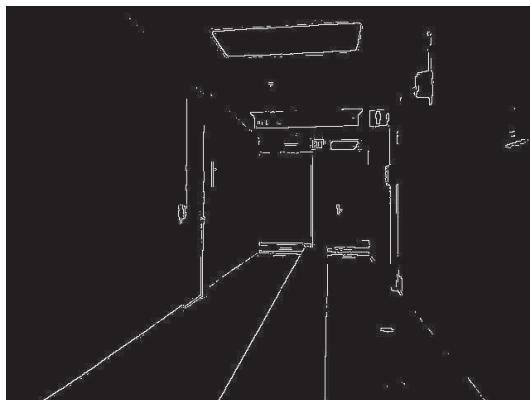
(b) Roberts
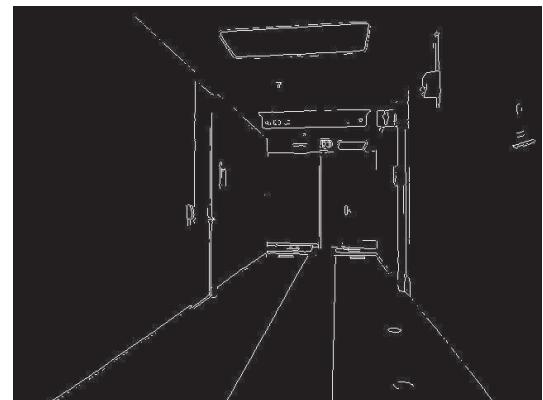
(c) Prewitt
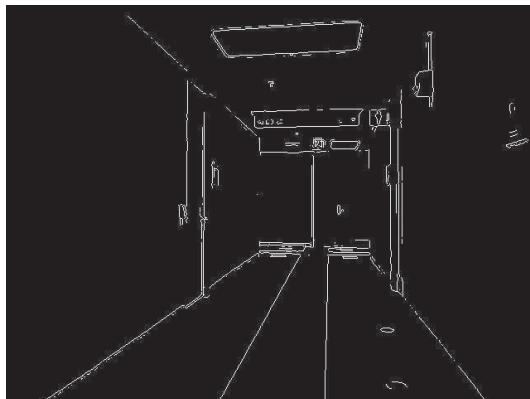
(d) Sobel
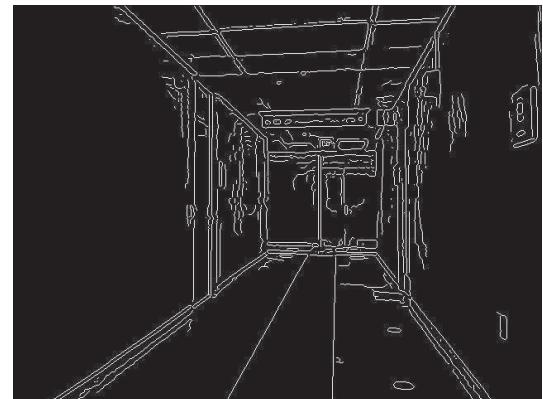
(e) Canny

Figure 3.2:   Edge detection comparison. The four indicated edge detection methods were used to produce subfigures (b) through (e) from the image in subfigure (a). The canny edge detection method returns more edges than the others.

The images used in this comparison are a different resolution than the ones used by Kessler et al., and the specific implementation of each method is likely different as well. These two factors account for the differing results obtained here. Though neither method is always faster than the other, the Hough transform was chosen as the line detection method for this research because both the Matlab and OpenCV software packages contain functions for implementing it within their respective libraries. With the line detection method selected, the vanishing point detection process can be developed.

3.1.4.3 Expressing Lines with Unit Normals. In order to find a vanishing point from the lines detected by the Hough transform, each line is first represented with a unit normal vector as explained in Section 2.5.2.4. This representation is obtained by removing the distortion from the pixel coordinates of each line's endpoints, converting the undistorted pixel coordinates to camera frame line-of-sight vectors with the $\boldsymbol{T}_{pix}^{c}$ matrix, and performing the cross product of each resulting pair. Each cross product is then divided by its magnitude to obtain a unit vector as shown in Equation (2.37).

3.1.4.4 Incorporating Prior Knowledge. Vanishing points are found by searching for places where many lines mutually intersect. Not every line in the image passes through, or even near, a particular vanishing point, so lines that are distant from where the vanishing point is likely to lie need not be included in the search for that vanishing point. To determine a likely area for finding a vanishing point, an initial prediction is made using Equation (3.28).

$$\mathbf{v}_{pred}^{c}(t_i) = \boldsymbol{C}_{b}^{c}\boldsymbol{C}_{\tilde{n}}^{b}(t_i^-)\mathbf{v}^{n} \tag{3.28}$$

Only lines that are near this prediction are used to find the measured vanishing point.

Since the magnitude of the dot product between a line and a point indicates how close the two are to one another, only lines whose dot product with the predicted

vanishing point has a magnitude less than a statistically-determined threshold are considered to correspond to the vanishing point of interest. The dot product of two unit vectors is equivalent to the cosine of the angle between them, and a point that lies on a particular line will be perpendicular to that line's unit normal vector. This means that, by definition, the angle between a line's unit normal and any point on the line is 90 degrees. Thresholding on the magnitude of the dot product then amounts to determining a maximum angular deviation from 90 degrees between the predicted vanishing point and each line's unit normal. The angular threshold chosen for this work is 5 degrees, which corresponds to a maximum dot product magnitude of 0.087. Figure 3.3 shows on an artificial edge image which lines are within 5 degrees of a particular point. The lines within the dot product threshold are called "support
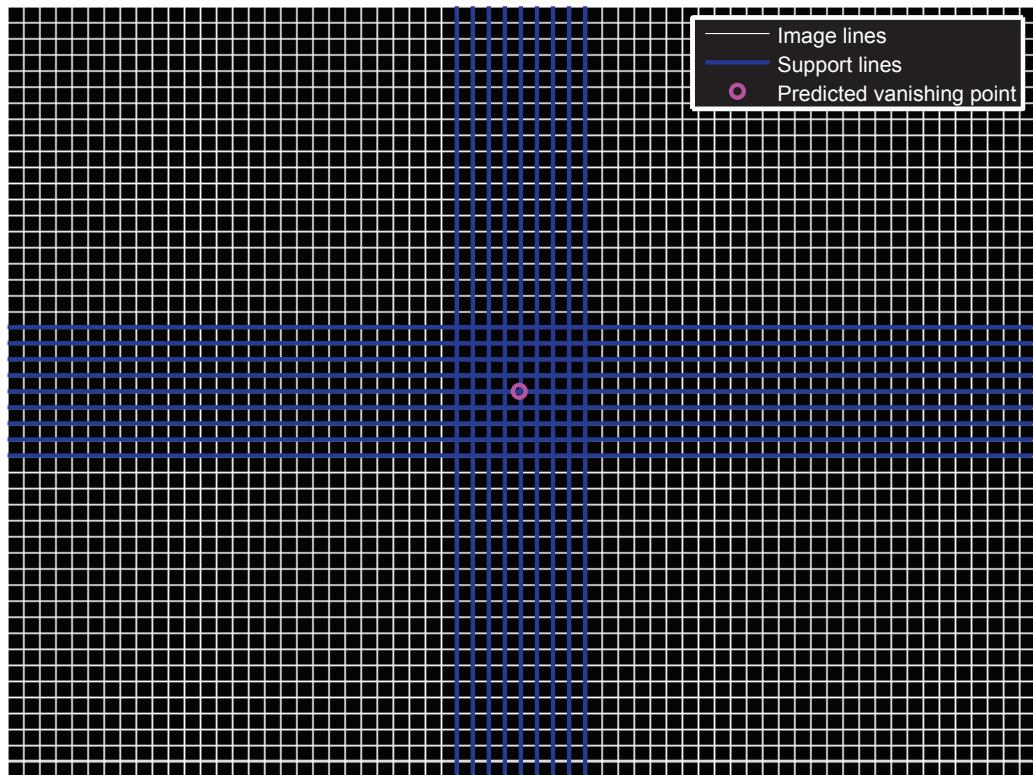

Figure 3.3:   Lines near a point. The lines shown in blue are within 5 degrees of the marked point.

lines" for that particular vanishing point. With the set of support lines determined, a vanishing point can be found from its contents.

The linearized measurement equation (3.25) and measurement prediction equation (3.28) both require that a body-to-camera frame DCM, $\boldsymbol{C}_b^c$, be known before they can be evaluated. No actual air vehicle was used to carry the test camera during this research, so the camera itself is treated as the vehicle. As described in Section 2.2, the x-axis of the vehicle body reference frame is typically oriented towards the direction of travel, which would be out the lens of the camera in this case. The z-axis of the camera reference frame is typically oriented out the lens as well. To maintain consistency with these conventions, the body-to-camera frame DCM simply performs a 90-degree rotation about the body-frame's y-axis. This DCM is shown in Equation (3.29).

$$\boldsymbol{C}_b^c = \begin{bmatrix} 0 & 0 & -1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \tag{3.29}$$

3.1.4.5 RANSAC for Vanishing Point Detection.   Recall that the RANSAC algorithm uses a random minimum subset of experimental data to build a model and then finds how much more of the data fit this model. After a statistically determined number of random subsets have been examined, the one with the largest consensus set is declared the best fit to the data and the model is refined using only members of this consensus set, i.e., inliers.

A vanishing point is a point where many lines mutually intersect. The smallest minimum subset to model such a point is the intersection of only two lines. Therefore, two of the support lines are selected at random, and their intersection calculated by evaluating the normalized cross product of their unit normal vectors as shown in Equation (2.38). This intersection is then compared to the complete set of support lines to find how many other lines pass near it. Once again, the dot product is used to make this comparison. The magnitude of the dot product between the initial intersection point and the remaining support lines is calculated and those below a threshold are added to the inliers of that consensus set. In this comparison, only lines

whose planar normals are within 0.5 degrees of 90 degrees from the intersection are considered inliers to that candidate intersection. This angular threshold corresponds to a maximum dot product magnitude of 0.0087. Figure 3.4 shows which lines in an artificial edge image are within 0.5 degrees of an initial candidate intersection.
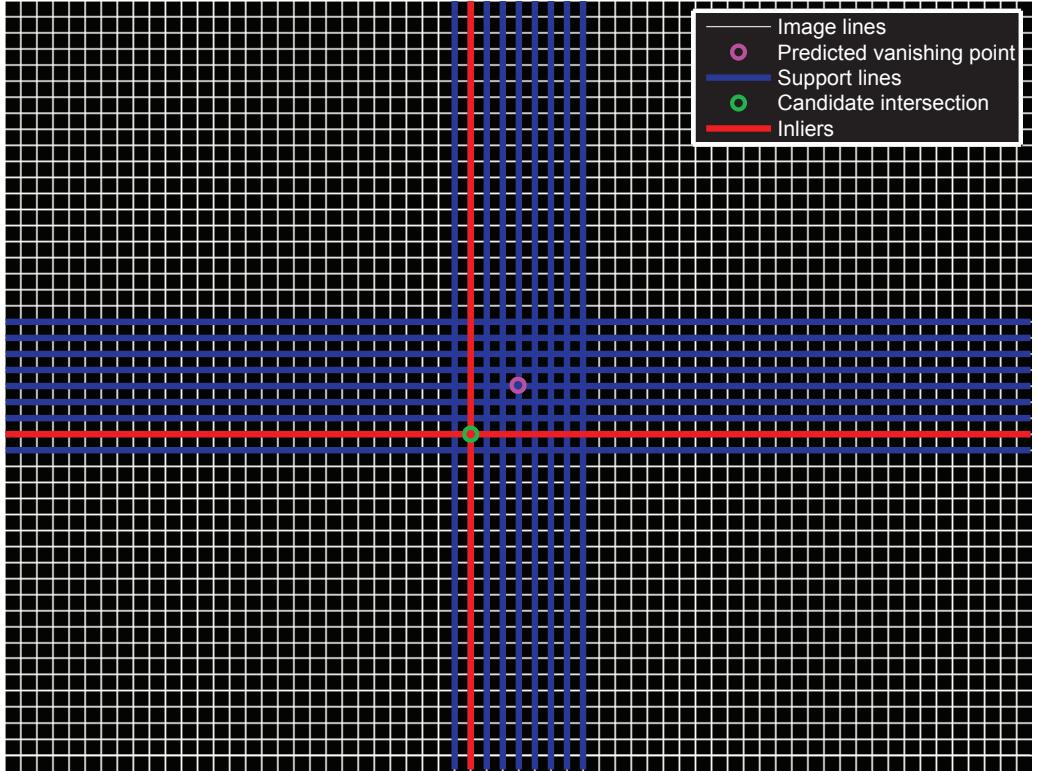
Figure 3.4:   Inlier lines. The lines shown in red are inliers to the intersection marked with the green circle.

Since the proportion of inliers is not known from the outset, the iterative approach for determining the minimum number of random pairings to examine described at the end of Section 2.5.3.2 must be used. An initial estimate of 1 is used for the proportion of outliers, ϵ. This results in an initial value of infinity for the minimum number of random pairings to try, but the number is quickly reduced as larger consensus sets are found.

In some instances, there may not be a single largest consensus set, i.e., multiple candidate intersections may have the same number of inliers. For such cases, a tie-

breaking criterion is necessary to determine the best consensus set. How close the inliers are to their corresponding candidate intersection provides this criterion. The set with the tightest grouping as determined by the sum of the dot products between each line and the candidate intersection is chosen as the better consensus set.

In other instances, the largest consensus set determined from the set of support lines may only contain the original two lines used to find the candidate intersection. In this case, the algorithm ends in failure and no vanishing point is declared in that particular direction for that image.

3.1.4.6 Vanishing Point Estimate Refinement. If a consensus set is found, the final step in finding the vanishing point is to find the point that best fits the consensus set of lines. This is accomplished using static optimization techniques. Such techniques begin with the definition of a cost function. The best estimate of the vanishing point from a group of lines is the point which minimizes the dot product with each of the lines in the consensus set. This can be expressed mathematically as:

$$J_k = \sum_{i=1}^{N_k} (\boldsymbol{l}_i^T \mathbf{v}_k)^2 \quad \forall \ k \in [1,3] \tag{3.30}$$

where N_k is the number of lines in the consensus set for the k-th vanishing point and J_k is the total cost associated with $\mathbf{v}_k$. Unfortunately, minimizing the cost function shown in Equation (3.30) will always lead to the trivial solution, $\mathbf{v}_k = \mathbf{0}_{3\times 1}$. While this solution does, in fact, minimize J_k, it does not reveal the direction of the vanishing point. The trivial solution can only be avoided by adding a constraint to the cost function through the use of a Lagrange multiplier, η. The constraint to be added in this case is to require that $\mathbf{v}_k$ must be of unit length. Equation (3.31) shows the new, constrained cost function.

$$J_k = \sum_{i=1}^{N_k} (\boldsymbol{l}_i^T \mathbf{v}_{k_x})^2 + \eta (\mathrm{v}_{k_x}^2 + \mathrm{v}_{k_y}^2 + \mathrm{v}_{k_z}^2 - 1) \quad \forall \ k \in [1,3] \tag{3.31}$$

The optimized solution is then found by implementing a gradient descent technique to solve for the values of $\mathbf{v}_k$ and η which satisfy:

$$\nabla_{\mathbf{v}_k,\eta} J_k = \mathbf{0}_{4\times 1} \quad \forall \ k \in [1,3] \tag{3.32}$$

The candidate intersection is used as the starting point for the iterative process of determining the value of $\mathbf{v}_k$ that minimizes J_k. This solution is the measurement realization that is used to update the Kalman filter.

Once a vanishing point is found, the inlier lines associated with it are removed from the total set of lines in the image before moving on to search for the next vanishing point. This is done to reduce the amount of extraneous information presented to the vanishing point detection process in subsequent directions and is valid because the only line which would legitimately pass through two vanishing points is a horizon line. Since the host vehicle is assumed to be operating indoors, the horizon will not be in the camera's view.

3.1.4.7 Direction Disambiguation. Every pair of lines in an image intersects in exactly two places, one 180 degrees offset from the other. This is evidenced by the effect of reversing the order of the cross product in Equation (2.38). Due to the random nature of the initial pairing of image lines performed by the RANSAC method, it is possible for the algorithm to converge on a vanishing point that is opposite the one being sought. To obtain the correct measurement under these circumstances and prevent degrading the overall attitude estimate, each measurement is compared with the measurement prediction to confirm that it is in the same hemisphere. If the component of the predicted vanishing point with the largest magnitude does not have the same sign as its corresponding component in the measured vanishing point, the measured vanishing point is flipped by multiplying it by a factor of -1. This ensures that the predicted and measured vanishing points are always in the same hemisphere.

3.1.4.8 Residual Monitoring. Occasionally, it is possible for the vanishing point detection algorithm to commit large errors. For instance, measurement noise may cause the camera's perception of a tight group of lines that are horizontally parallel in the world to appear to cross on the image plane. In such a case, the z-component of the horizontal vanishing point will be found to be nearly equal to 1 when the true vanishing point has a z-component nearly equal to 0. This presents a very large measurement residual and the measurement should be ignored. The diagonals of the residual covariance matrix, $\boldsymbol{H}_k(t_i)\boldsymbol{P}_{\psi\psi}(t_i^-)\boldsymbol{H}_k^T(t_i) + \boldsymbol{R}_k$, can be used to filter out such erroneous measurements. A measurement that yields a residual whose magnitude in any of its components is larger than 3-σ can be safely ignored with 99.6% probability. Stated another way, discarding measurements which lie further than 3-σ from the prediction will only eliminate a valid measurement 0.4% of the time.

3.2 Experimental Methods

This section presents the experimental methods used to evaluate the attitude estimation technique described in Section 3.1. The topics presented here include a description of the test equipment and an outline of the procedures used to acquire test data.

3.2.1 Equipment. A variety of test equipment was used to collect data for evaluating the attitude estimation method presented in Section 3.1. The system under test consists primarily of the same models of MEMS IMU and camera with which the proposed host vehicle is equipped. These sensors and their associated circuitry are housed inside an approximately 80 mm × 60 mm × 160 mm plastic box as shown in Figure 3.5. This sensor box serves as a mock-up of the quadrotor for which the attitude estimation algorithm is intended. Additionally, a tactical-grade IMU was used to provide more accurate attitude data than the estimates provided by the Kalman filter for later comparison.
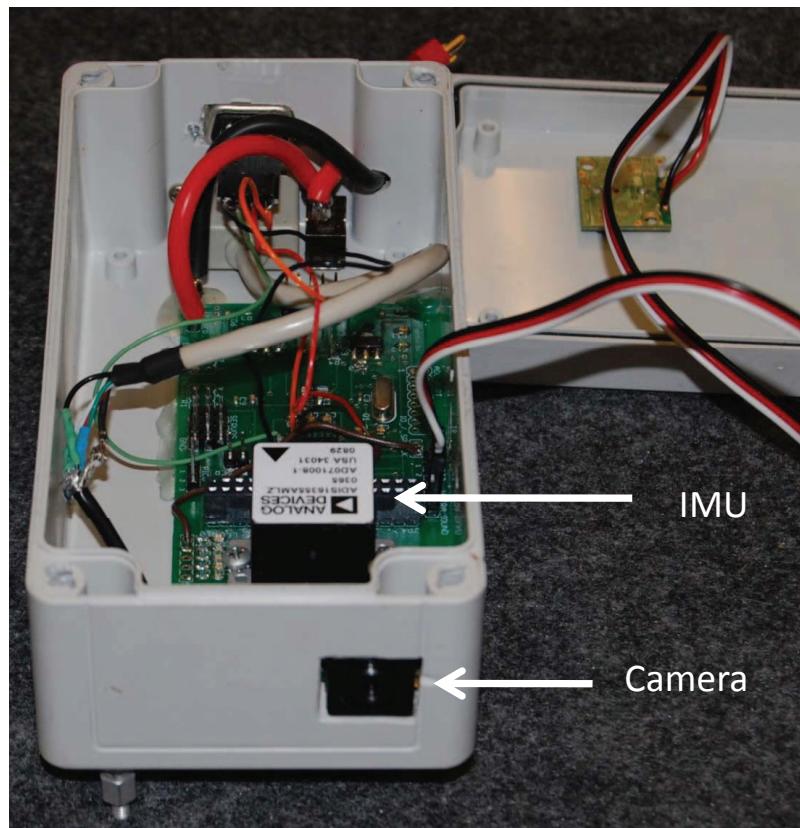
Figure 3.5:   System under test. The system under test consists of the IMU and camera shown here housed inside a plastic box.

3.2.1.1 MEMS IMU. The MEMS IMU used onboard the proposed host vehicle is the ADIS 16355 produced by Analog Devices, Inc. It is a complete triple-axis inertial unit contained inside an approximately 23 mm × 23 mm × 23 mm cube. Its small size, light weight and low power requirement make it an attractive option for applications such as the one in this thesis.

3.2.1.2 Tactical-Grade IMU. The tactical-grade IMU used to compare performance with the attitude estimation technique is the HG1700-AG58 manufactured by Honeywell. This is a much higher performance device than the MEMS IMU, as it is equipped with ring laser gyroscopes. While it provides much more accurate angular rate measurements than the MEMS IMU, it is also much larger, heavier and requires significantly more power. The specifications of both the MEMS and tactical IMUs as published by their respective manufacturers are shown in Table 3.2 for side-by-side comparison.

Table 3.2: IMU specifications

Parameter (units)	MEMS IMU	Tactical IMU
Sample rate (Hz)	≤ 819	100
Input range (deg/s)	±300	±1000
Gyro rate bias (deg/hr)	54	1
Angular random walk (deg/$\sqrt{\text{hr}}$)	4.2	0.125
Dimensions (mm)	23 × 23 × 23	168 × 196 × 146
Weight (g)	16	4500
Power required (W)	≤ 0.3	~8

3.2.1.3 Camera. The camera used onboard the proposed host vehicle is the Webcam Pro 9000 produced by Logitech. It is equipped with a two megapixel sensing array and the associated optics. There is also a dynamic autofocus feature, but for this application, the autofocus was disabled. While disabling the autofocus sometimes results in blurred images, it ensures that the focal length remains constant. The constant focal length facilitates the use of a single set of calibration parameters

for the image processing tasks. The camera was used to capture 640 × 480 resolution grayscale images for the optical-aiding portion of the Kalman filter.

3.2.1.4 Peripherals.   In addition to the primary test equipment already described, various pieces of secondary equipment were also requisite in performing this research. Power to both the tactical IMU and inertial-optical sensor box was provided by a standard 12V car battery connected to an 800-Watt power inverter. Novatel, Inc.'s synchronized position, attitude and navigation (SPAN™) combined global navigation satellite system/inertial navigation system (GNSS/INS) receiver was used to interface with the tactical IMU. Also, two laptop computers were used to record the data generated by the two sensor platforms. A wheeled cart enabled mobility of the entire configuration for dynamic test events. Lastly, pieces of 8020 aluminum were used to construct a rigid frame to mount the sensor box and tactical IMU together and also to provide a structure from which this joint apparatus could be suspended. The combined testing rig is shown in Figure 3.6.

3.2.2 Procedures.   The experimental procedures for collecting the data used to evaluate the attitude estimation technique consisted primarily of two main tasks– camera calibration and motion profile development. These tasks were performed to enable an evaluation of the attitude estimation method.

3.2.2.1 Camera Calibration.   Before any of the steps beyond line detection in the image aiding process can be implemented, an intrinsic camera matrix and optical distortion parameters must be known. These are found by performing a calibration of the test camera. Multiple images of a camera calibration board were captured at varying range and orientation and then processed using Bouget's Camera Calibration Toolbox for Matlab [3] to extract the desired parameters. The results are shown in Table 3.3.
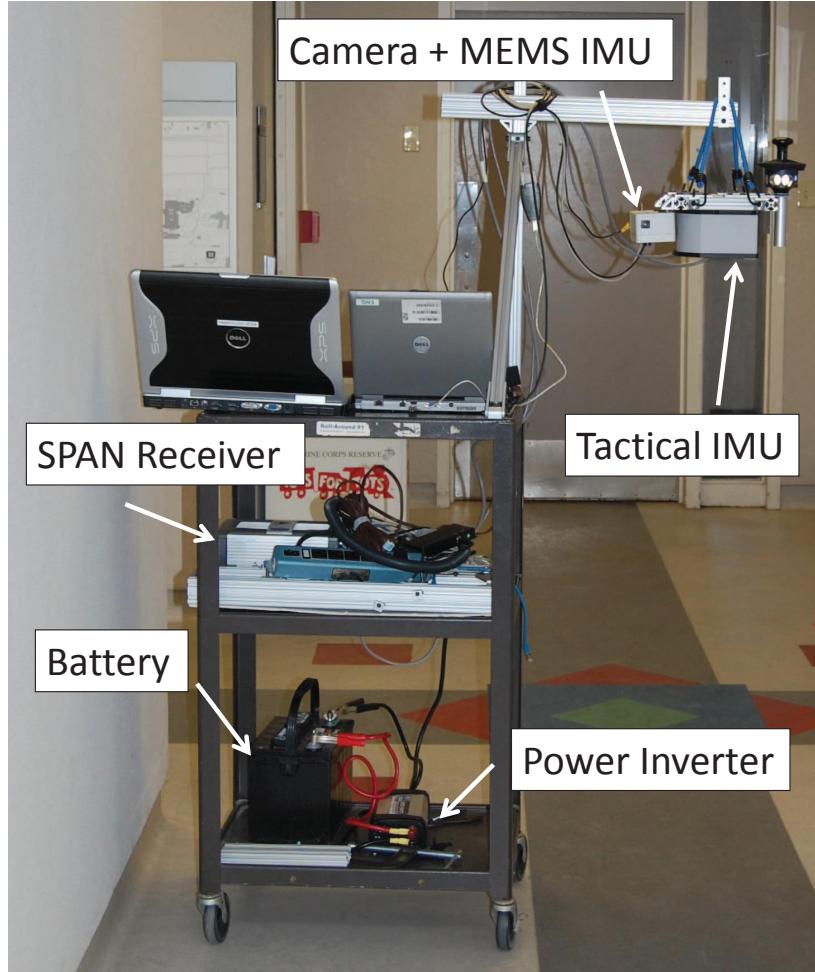
Figure 3.6:   Test rig. The equipment shown here were used to collect data for evaluating the attitude estimation method described in this thesis.

Table 3.3:   Camera calibration parameters. Uncertainties express 3-σ boundaries.

Parameter (Units)	Value
Focal length (pixels)	$[528.6, \ 528.6] \pm [0.3, \ 0.3]$
Principal point (pixels)	$[307.8, \ 224.1] \pm [0.5, \ 0.5]$
Skew factor	$3.5 \ \times 10^{-4} \pm 1.6 \times 10^{-4}$
Radial distortion coefficients	$[0.043, \ -0.17, \ 0.075] \pm [0.0026, \ 0.01, \ 0.012]$
Tangential distortion coefficients	$[-9 \times 10^{-5}, \ 7.3 \times 10^{-4}] \pm [2.3 \times 10^{-4}, \ 2.1 \times 10^{-4}]$

3.2.2.2   Motion Profile.   Since the attitude estimation algorithm is intended for use on a flying vehicle, a test profile was developed to mimic the motion of flight. The combined sensor box and HG1700 IMU apparatus was used to represent the vehicle. A rectangle was marked on the floor with electrician's tape outlining the apparatus's starting position so that it could be returned to roughly the same orientation for each of 15 sample runs. Approximately 60 seconds of static data were collected at the beginning of each sample run before the mock vehicle was lifted off the floor and hung from the wheeled cart as shown in Figure 3.6. Then, the cart was pushed through one hallway towards an intersection with another, clockwise around the corner, and through the second hallway, stopping approximately 4.5 meters shy of its end. This approximate trajectory is depicted in Figure 3.7. After any swinging motion in the mock vehicle had subsided, another 60 seconds of static data were collected at the end of each run.
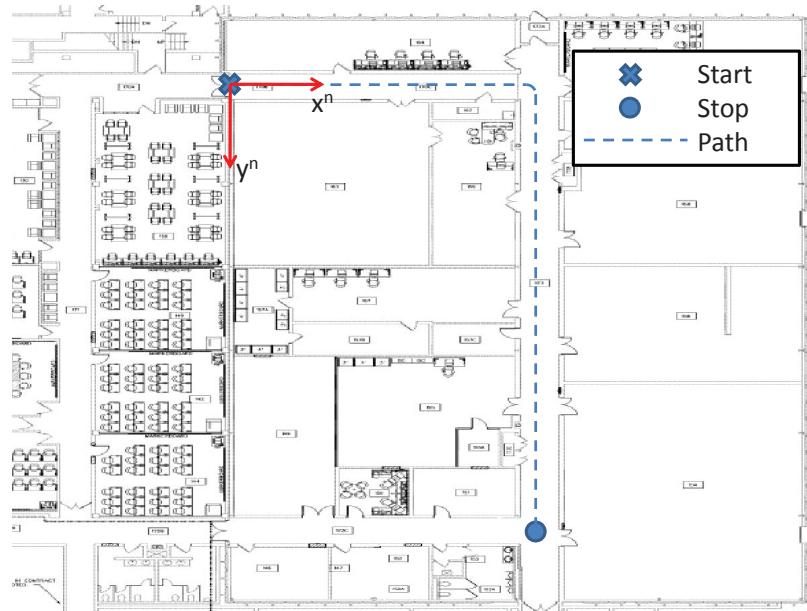
Figure 3.7:   Pathway through halls. The test apparatus began at the origin of the Earth-fixed navigation frame and approximately followed the indicated path through the hallways.

3.3 Summary

This chapter has discussed the development of the attitude estimation algorithm and the experimental procedures used to evaluate it. With these tasks performed, the test data can be processed and the results presented and analyzed in the next chapter.

IV. Results and Analysis

This chapter discusses the results of implementing the coupled vanishing point and inertial attitude estimation technique on the data that were collected during experimentation. The methods used to process the experimental data are presented in Section 4.1, the results of the vanishing point detection procedure are presented and analyzed in Section 4.2 and the attitude estimates are presented and discussed in Section 4.3.

4.1 Data Processing

After the experiments described in Chapter III were performed, the data collected were postprocessed using The Mathworks, Inc.'s Matlab software. The data processing procedures include calculating the gyro biases, generating attitude profiles from the unaided MEMS inertial, vision-aided MEMS inertial, and unaided tactical inertial data, and determining the errors in the estimated attitude profiles.

4.1.1 Gyro Bias Calculation. Equation (2.13) expresses the gyroscope measurements as the sum of the true relative rotation rate between the inertial and body reference frames, the vector of gyro biases, and zero-mean, white, Gaussian noise. During the initial static portion of each data run, the test apparatus is known to be at rest. Thus, because the Earth's rotation rate can be neglected, the gyro biases can be directly estimated. The initial static portion of each run has a total duration of approximately 60 seconds, in which time the Earth will have rotated approximately 0.25 degrees. This total rotation is within the expected margin of error for the estimation technique, so neglecting the Earth rate is justified. Assuming an Earth turn rate equal to zero and a stationary inertial sensor means that the true rotation rate can also be assumed to equal zero. Therefore, the bias is the only factor that will contribute to the mean of the gyro measurements during the initial stationary portion of each data run. As was discussed in Section 2.3.2, the bias will be treated as a fixed, deterministic quantity for the duration of a single test run. This means that the vector of gyro biases for a particular run can be determined by calculating the

69

mean of the first minute's worth of inertial measurements.

$$\boldsymbol{b}^b = \frac{1}{60} \int_0^{60} \boldsymbol{\omega}_{ib_m}^b(t)dt \tag{4.1}$$

Figure 4.1 shows the biases that were determined from each of the 15 test runs. The largest biases are in the measured rotation rate about the mock vehicle's pitch, i.e., the body frame's y, axis. These correspond to an average measured rotation rate of approximately 4 degrees per second obtained while the IMU was held stationary. Clearly, these biases can degrade the overall attitude estimate if they are not compensated for.
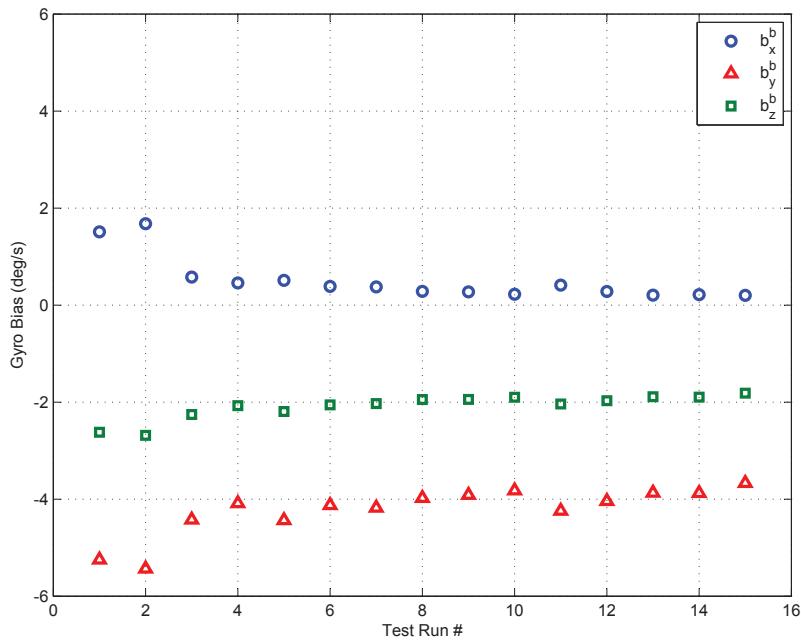

Figure 4.1: Gyro biases. The biases in each of the 3 MEMS gyros calculated from each of the 15 test runs are shown here.

4.1.2 Unaided Inertial Attitude Profile Generation. Once the gyro biases are known, unaided inertial attitude profiles can be calculated. Beginning from an initial attitude estimate, unaided attitude profiles based on measurements taken from either the MEMS or tactical IMU are generated using Equation (3.7). The attitude profiles originating from the tactical inertial data are the most accurate available for

comparison with attitude estimates from other sources. The HG1700 IMU used to obtain these measurements has a drift rate of only 1 degree per hour, and a test run has a duration of only approximately 3.5 minutes. In this time, the tactical inertial solution will have drifted only 0.06 degrees. Therefore, the tactical inertial attitudes will be assumed to be equivalent to truth for the purpose of calculating attitude errors. An example of one unaided MEMS inertial profile and the corresponding tactical inertial profile are shown in Figure 4.2. These attitude profiles are expressed in terms of Euler angles. Note the drift in the unaided MEMS inertial profile. It is this drift that we wish to constrain through the use of the vision-aiding Kalman filter.

4.1.3 Error Calculation. A method for determining attitude errors is necessary to be able to evaluate the accuracy of any of the attitude profiles developed during this research. Equations (3.1) and (3.2) can be combined to facilitate error calculation as:

$$\boldsymbol{C}_b^n = e^{\boldsymbol{\psi} \times} \boldsymbol{C}_b^{\tilde{n}} \tag{4.2}$$

Solving this equation for the vector of error angles, $\boldsymbol{\psi}$, yields:

$$\boldsymbol{\psi} \times = \ln \left(\boldsymbol{C}_b^n \boldsymbol{C}_{\tilde{n}}^b \right) \tag{4.3}$$

The body-to-navigation frame DCM calculated from the tactical inertial data is substituted for $\boldsymbol{C}_b^n$ before calculating the matrix logarithm on the right side of Equation (4.3). Finally, collapsing the skew-symmetric matrix gives the vector of errors, $\boldsymbol{\psi}$.

Figure 4.3 shows the results of applying this error calculation method to the unaided inertial profile shown in Figure 4.2. Again, the unbounded drift in the unaided attitude profile is clearly evident. The abrupt change in all three components of the error vector at about 100 seconds corresponds to when the simulated vehicle turned the corner from one hallway to another. This phenomenon is present in all of the 15 test runs.
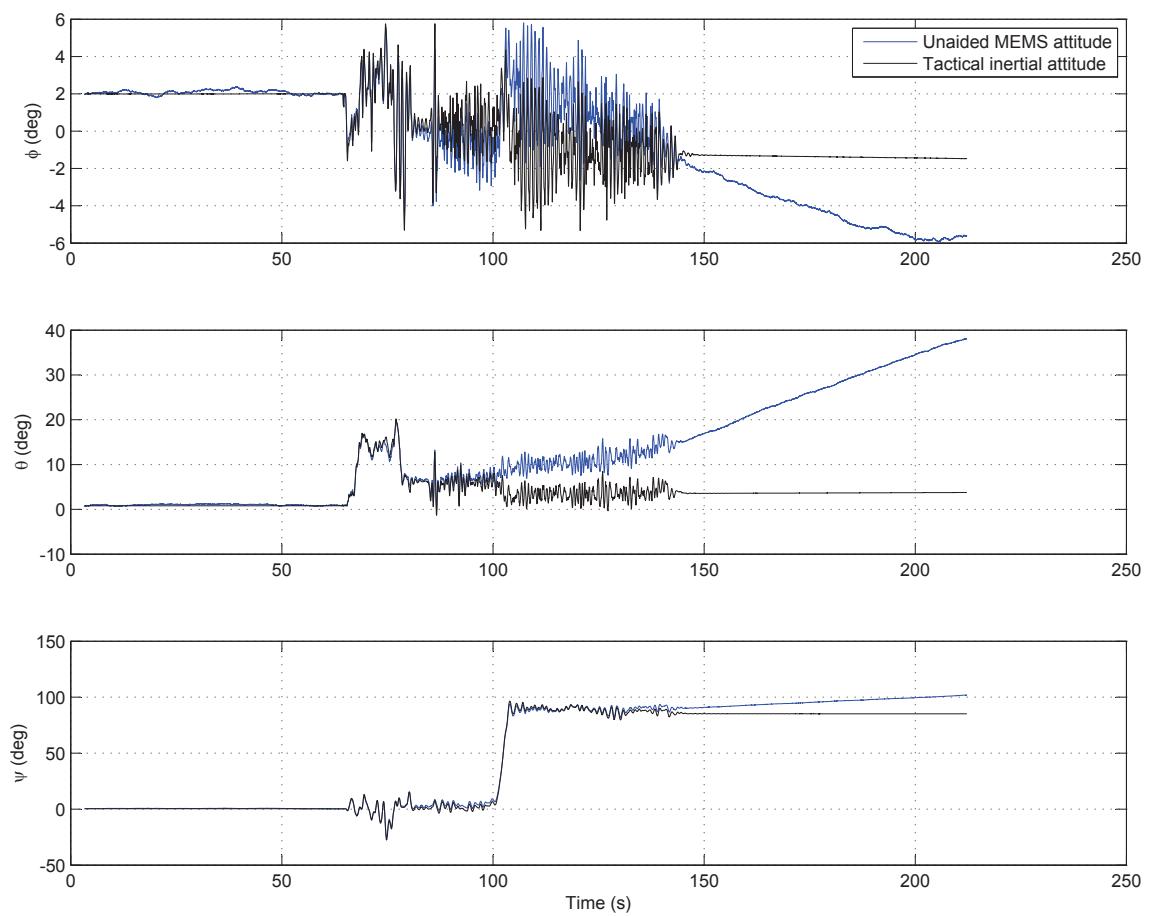
Figure 4.2: Unaided inertial attitude profile. The unaided MEMS inertial attitude profile shown here drifts unbounded due to additive inertial measurement noise.
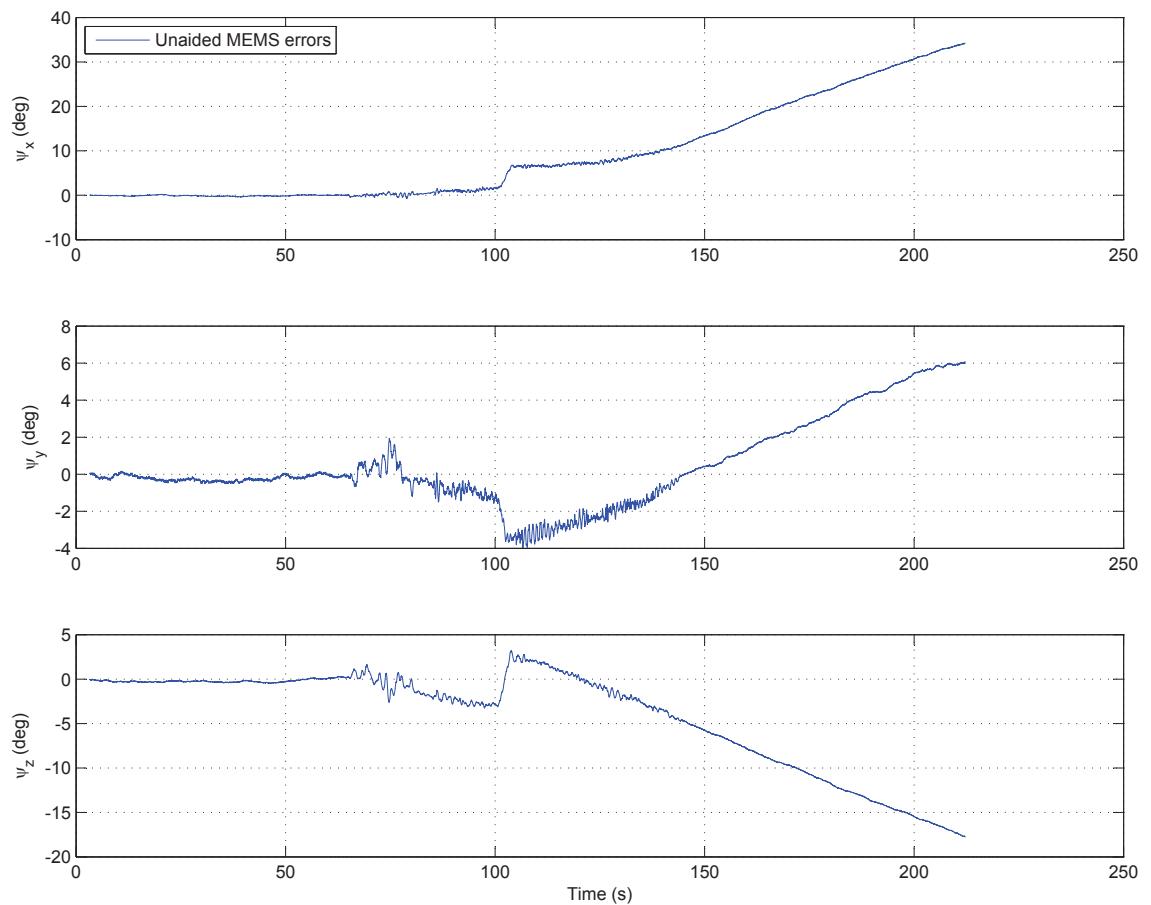
Figure 4.3: Unaided MEMS inertial errors. The errors in the unaided MEMS inertial profile shown in Figure 4.2 are displayed here.

4.1.4   Process Noise Characterization.   With an error calculation method in place, the discrete-time process noise strength, $\boldsymbol{Q}_d$, can be determined. This is accomplished by performing an analysis of the ensemble of errors in the complete set of unaided MEMS inertial attitude solutions. The method described in Section 4.1.3 is used to find the errors in the unaided MEMS inertial solutions from each of the test runs, and the resulting ensemble of errors is used to calculate standard deviations for the full set of test data. Figure 4.4 shows the ensemble of unaided MEMS inertial errors for all 3 gyroscopes and all 15 data runs along with the calculated ensemble standard deviations. The time segment shown begins after the initial 60 second static period over which the gyro measurements have been averaged to determine the gyro biases.
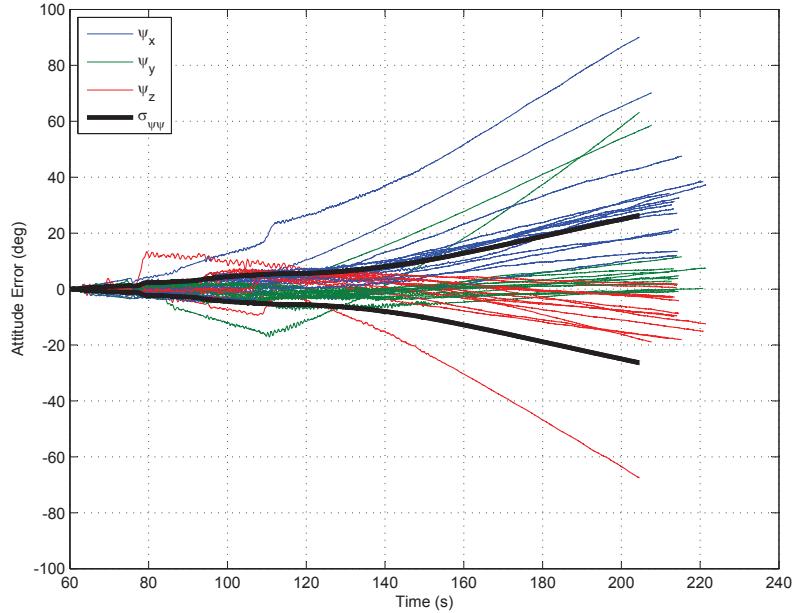
Figure 4.4:   Unaided MEMS inertial ensemble errors. The errors present in the unaided MEMS inertial solutions in all 3 axes from all 15 test runs and the calculated ensemble standard deviations are shown here.

The process noise strength can be determined by finding a linear fit to the ensemble variances. The first-order coefficient of a linear fit to these variances reveals the rate at which the uncertainty in the attitude estimates increases when the MEMS inertial data are unaided. The linear fit to the unaided MEMS inertial ensemble

variances from 60 to 120 seconds is shown in Figure 4.5. The variances over this interval increase at a rate of approximately 0.63 deg²/s. Multiplying this rate by the time-step between MEMS inertial measurements (4.9 milliseconds) yields the following value for the discrete-time process noise strength:

$$\boldsymbol{Q}_d = \begin{bmatrix} 3.1 \times 10^{-3} & 0 & 0 \\ 0 & 3.1 \times 10^{-3} & 0 \\ 0 & 0 & 3.1 \times 10^{-3} \end{bmatrix} \deg^2 \tag{4.4}$$
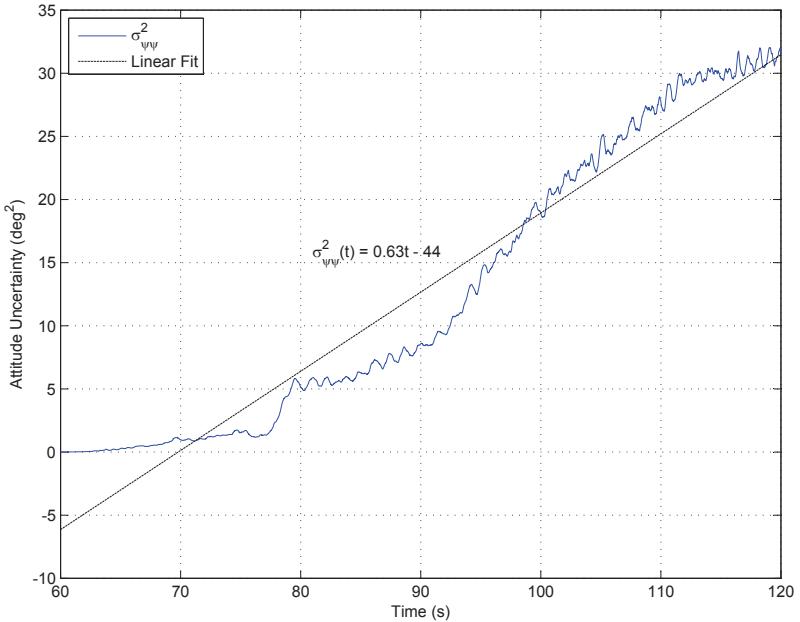
Figure 4.5: Unaided MEMS inertial variances. The process noise strength is determined from the first-order coefficient of the linear fit to the unaided inertial error variances.

4.2 Vanishing Point Detection

With the gyro biases and process noise strength determined, the attitude estimation technique described in Section 3.1 can be implemented on the data that were collected during the test runs. The vanishing points found in the images captured by the camera will be used to constrain the drift in the MEMS inertial solutions. There-

fore, the effectiveness of the vanishing point detection algorithm is discussed in this section before presenting the vision-aided attitude profiles and errors in the section that follows.

Figure 4.6 shows an image that was captured approximately 82 seconds into one of the test runs. In this example, all three vanishing points have been identified, and lines which have been declared by the RANSAC algorithm to be inliers to any of the three principal vanishing directions are highlighted. Note that some lines returned by the Hough transform have been rejected as outliers. Also, the predicted vanishing point in the direction down the hall and the corresponding vanishing point found in the image are close to one another, resulting in a small measurement residual.
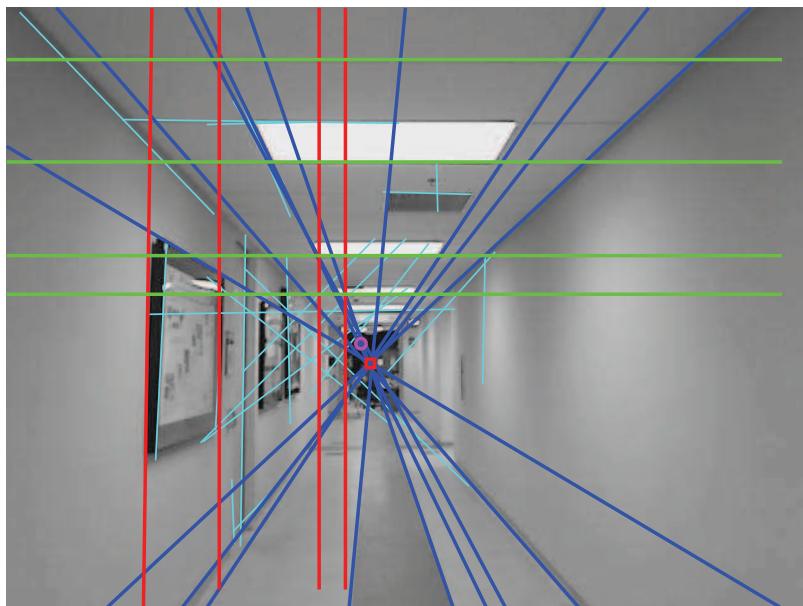

Figure 4.6:   Vanishing points found in a sample image. All three vanishing points have been identified in this image. The magenta circle is where $\mathbf{v}_1$ was predicted to be, and the red square is where $\mathbf{v}_1$ was found. The blue lines are inliers to $\mathbf{v}_1$, the green lines are inliers to $\mathbf{v}_2$, the red lines are inliers to $\mathbf{v}_3$, and the cyan lines are outliers.

4.2.1   Measurement Susceptibility to Noise.   Unfortunately, the RANSAC vanishing point detection scheme does not always return the type of result shown in Figure 4.6. An example of an image in which $\mathbf{v}_2$ is found to be only a few degrees from

$\mathbf{v}_1$ is given in Figure 4.7. Of course, $\mathbf{v}_2$ actually lies in a direction nearly parallel to the image plane. The three lines identified as inliers to this spurious measurement pass close enough to the prediction of $\mathbf{v}_2$ that they meet the criteria for being classified as support lines. However, the glancing angle at which the camera views the horizontal stripe on the floor introduces significant uncertainty into the locations of the lines defined by the edges of the stripe. This imprecision is what makes the lines appear to intersect on the image plane.
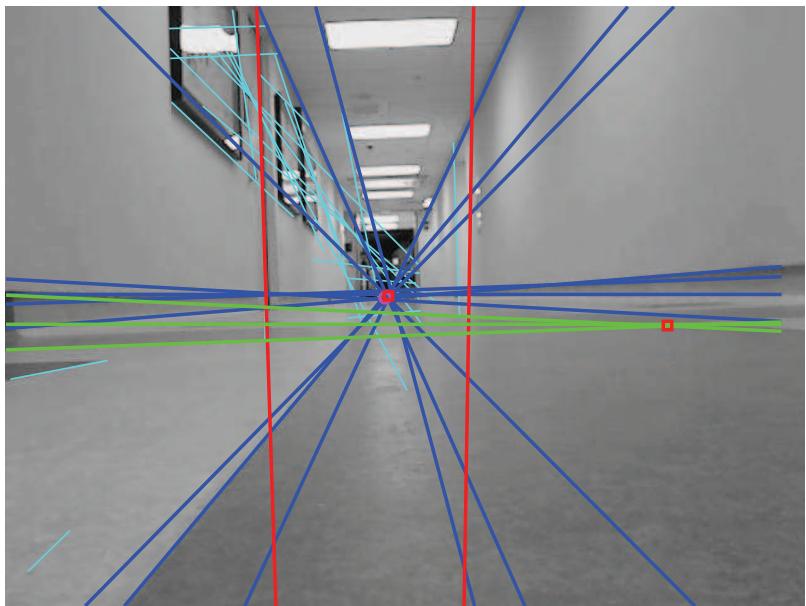

Figure 4.7: Spurious vanishing point measurement. The intersection of the inliers to $\mathbf{v}_2$ is found on the image plane when the true vanishing point clearly lies far from this measurement.

4.2.2 Measurement Residuals. The sequence of residuals in the Kalman filter shows how close the measurements are to their respective predictions. This information can reveal how often spurious measurements such as the one discussed in Section 4.2.1 occur. The residual monitoring process described in Section 3.1.4.8 prevents these spurious measurements from unduly influencing the attitude estimates. Recall that if any component of a residual is found to lie outside of three standard deviations from zero, the corresponding measurement is ignored by the Kalman filter and treated as if no measurement were obtained for that vanishing point.

Figure 4.8 shows the sequences of residuals and their corresponding 3-σ boundaries from one of the test runs. The numeral subscript on a residual indicates the vanishing point to which the residual corresponds, e.g., $\delta \boldsymbol{z}_1$ is the residual corresponding to the measurement of $\mathbf{v}_1$. In the example shown in the figure, spurious measurements of $\mathbf{v}_2$ frequently occur at the beginning of the test run while the mock vehicle is resting on the ground. The intersection of the green lines in Figure 4.7 is an example of one such measurement. After the mock vehicle has been lifted off the floor at approximately 65 seconds into the test run, the residuals for $\mathbf{v}_2$ are much smaller.

Another observation that can be made from the residuals is with respect to the uncertainty in the z-components of the vanishing point measurements. When the camera is looking away from a particular vanishing point, there is much greater uncertainty in the z-component of the corresponding measurement than when the camera is pointed toward the vanishing point. Looking again at Figure 4.8, there is much greater variability in the z-component of $\delta \boldsymbol{z}_2$ before the mock vehicle turns the corner approximately 100 seconds into the run than after. After the corner, the camera is peering in the direction of $\mathbf{v}_2$, and the fluctuations in the z-component of $\delta \boldsymbol{z}_2$ are no longer observed. Now, however, the z-component of $\delta \boldsymbol{z}_1$ varies much more widely than it had before. The z-component of $\delta \boldsymbol{z}_3$ exhibits a greater degree of variability than the x or y-components for the duration of the test run, because the camera is never looking in the direction of $\mathbf{v}_3$, i.e., straight up or straight down. These observations provide evidence that when the camera is pointed toward a particular vanishing direction, the uncertainty in the z-component of the corresponding vanishing point is much smaller than the z-components of the other two vanishing points. The 3-σ boundaries may seem large somewhat large, particularly over portions in the test run where the residuals are consistently small. However, it will be shown that the filter's calculated uncertainties closely match the test runs' ensemble variances.

Lastly, the residuals show that in this test run there were few reliable updates to the roll estimate while the mock vehicle was resting on the floor. When the camera is peering in the direction of $\mathbf{v}_1$, the roll axis is not observable using measurements
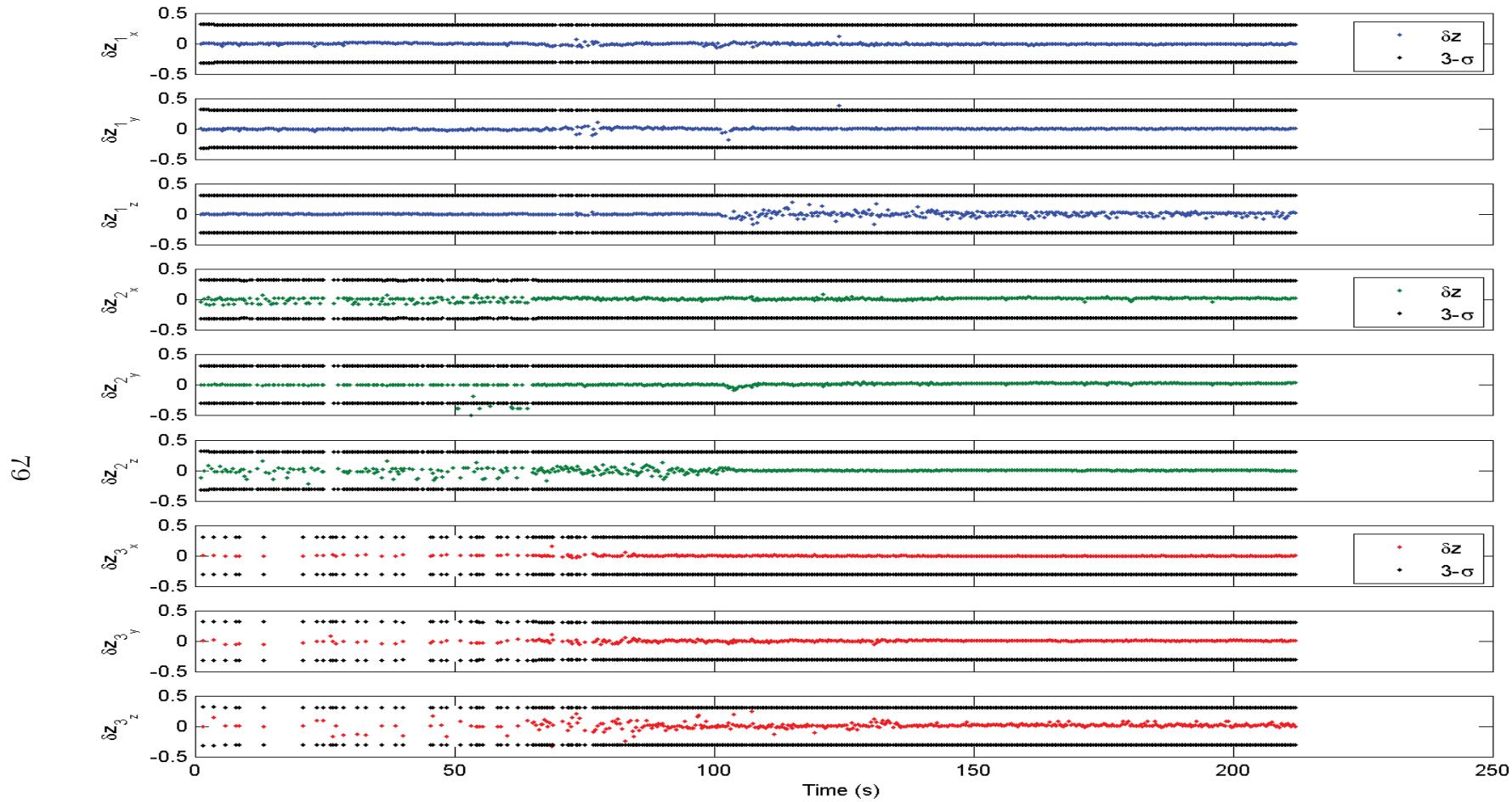
Figure 4.8: Measurement residuals. The residuals in the x, y and z-components of all three vanishing point measurements for a single test run are shown here with their respective 3-σ covariance boundaries. Any measurements outside of the 3-σ threshold are rejected and are not incorporated into the attitude estimates.

of that vanishing point, so those measurements do not influence the roll estimates. This means that information influencing the roll axis must come from either $\mathbf{v}_2$ or $\mathbf{v}_3$. However, many of the measurements of $\mathbf{v}_2$ were discarded because they exhibited large residuals, and few measurements of $\mathbf{v}_3$ were obtained. Nonetheless, even under these adverse conditions, the roll estimates will be shown to remain within a few degrees of the tactical inertial roll profile.

4.3   Kalman Filter Attitude Estimates

Now that the vanishing point detection method has been evaluated, the attitude profiles obtained by coupling the MEMS inertial data and vanishing point measurements will be presented and discussed. Examples of an attitude profile generated by the Kalman filter and the corresponding attitude profiles calculated from the tactical and unaided MEMS inertial measurements are shown in Figure 4.9. As was discussed in Section 4.2.2, the roll estimate is degraded during the initial static portion of the test run due to infrequent and inaccurate measurements of vanishing points $\mathbf{v}_2$ and $\mathbf{v}_3$. However, in the long term, the attitude estimates in all three axes are significantly more accurate than the unaided MEMS inertial estimates.

Performing the error calculation procedure discussed in Section 4.1.3 on both the unaided MEMS inertial and Kalman filter data from this test run yields the results shown in Figure 4.10. Again, the Kalman filter provides greater accuracy and stability in the long term. However, another phenomenon can also be seen in these error profiles. There is an overall bias in each of the Kalman filter's attitude estimates about which the solutions vary. This bias will be discussed in greater detail in Section 4.3.2.

4.3.1   Kalman Filter Ensemble Errors.   One test run alone cannot characterize the nature of the errors in the Kalman filter's attitude solutions. To help in understanding the filter's performance, the ensemble of errors in the Kalman filter's attitude estimates for all fifteen test runs and the corresponding means and standard
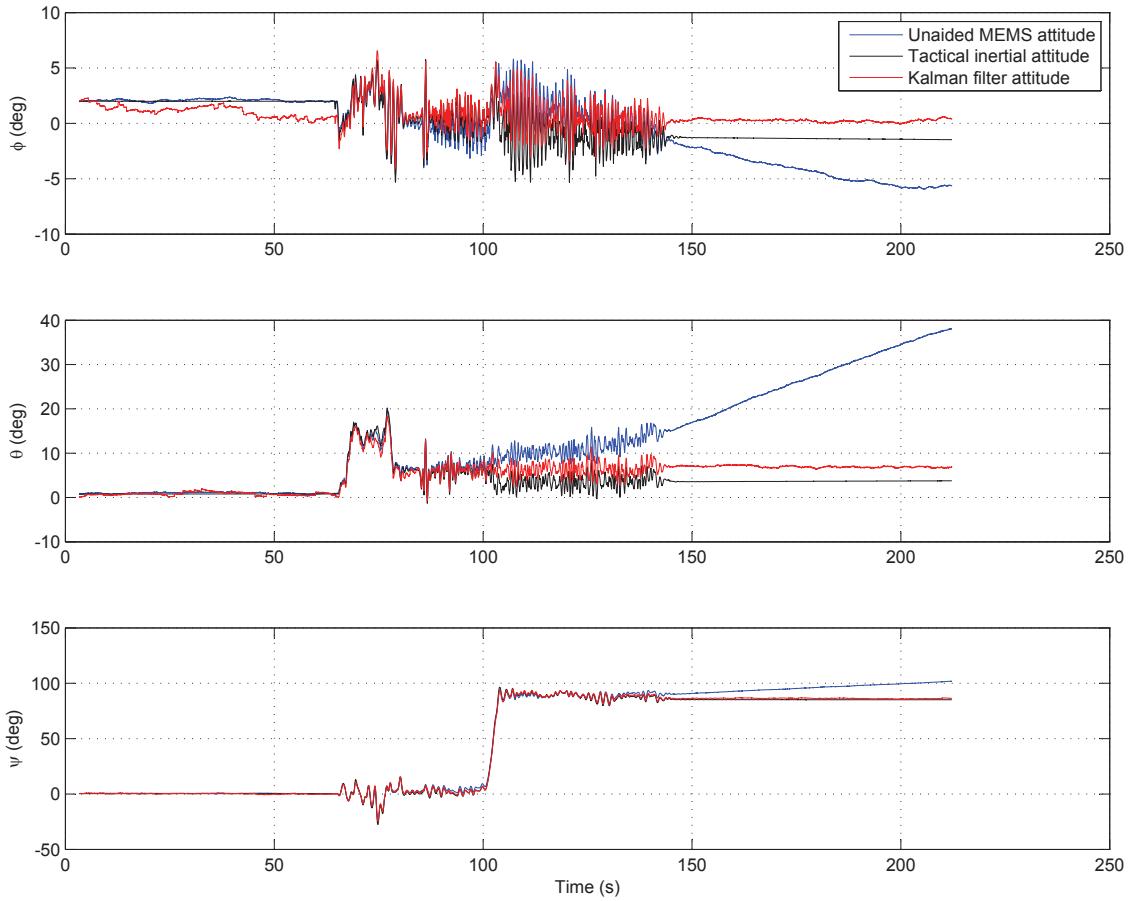
Figure 4.9: Kalman filter and unaided MEMS inertial attitude profiles from 1 run. The attitude profiles calculated from the Kalman filter, and both tactical and MEMS raw inertial data are shown here. Note the unbounded long-term drift present in the attitude profiles from the unaided MEMS inertial solution has been constrained in the Kalman filter attitude solution.
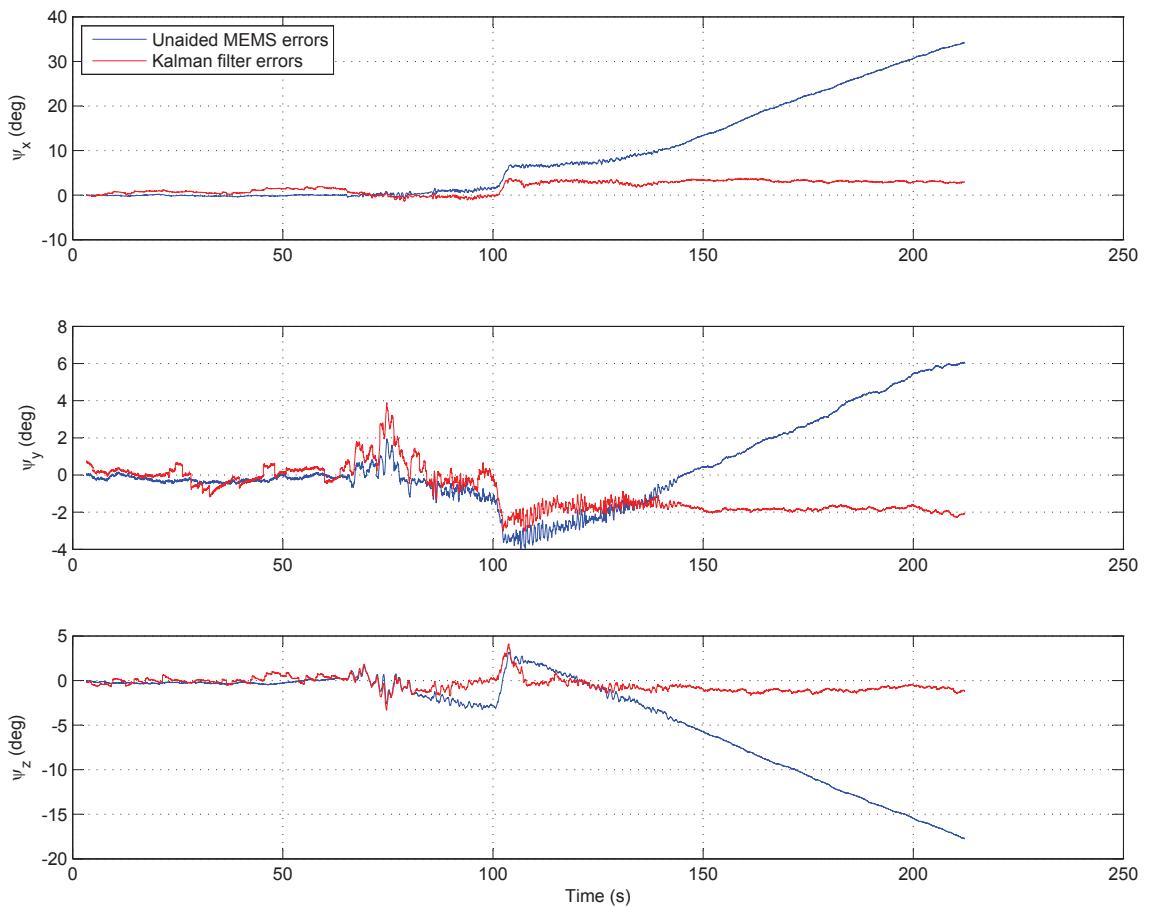
Figure 4.10:   Kalman filter and unaided MEMS inertial errors from 1 run. In the long term, the errors in the Kalman filter solution are much smaller than the errors in the unaided MEMS inertial solution.

deviations are shown in Figure 4.11. Comparing the ensemble errors in the Kalman filter's attitude solution shown in Figure 4.11 with the unaided MEMS inertial ensemble errors shown in Figure 4.4 shows that the long-term improvement exhibited by the single test case discussed earlier is present in all of the test runs. The unaided MEMS solutions have a standard deviation of approximately 26.4 degrees after 3.5 minutes which will only continue to grow over time. The vision-aided solutions, on the other hand, have standard deviations of approximately 1.5 degrees in the pitch and roll axes, and approximately 0.9 degrees in the yaw axis which are stable long-term. Respectively, these represent 94% and 96% reductions in the attitude uncertainties. Furthermore, the long-term stability of the Kalman filter's attitude estimates has entirely eliminated the unaided MEMS inertial solutions' drift rate of 0.63 $\deg^2$/s.

Another interesting comparison to make is between the uncertainties calculated by the Kalman filter and the ensemble standard deviations. Both of these sequences are shown together in Figure 4.12 to facilitate this comparison. The Kalman filter uncertainty profile from only one test case is displayed, for clarity, but the uncertainties from the others converge to nearly the same steady state as the one shown. The standard deviation calculated by the Kalman filter matches the ensemble standard deviation within one degree for most of the time segment with a peak difference of about 2 degrees observed in the yaw axis.

While the level of accuracy provided by the Kalman filter demonstrates a vast improvement over the unaided inertial solution, it would likely only partially meet the requirements for indoor flight. An inner control loop for a rotary vehicle will require accuracy to within a degree or two in the roll and pitch axes to control lateral motion. The combined effects of the 1.5 degree standard deviation in the errors and overall bias do not provide for a solution accurate to within two degrees consistently. However, there would be a larger tolerance for errors in the yaw axis, as precise heading is less important than relative position when operating indoors. Also, the Kalman filter's yaw solution has been shown to be more accurate than the pitch and roll solutions.
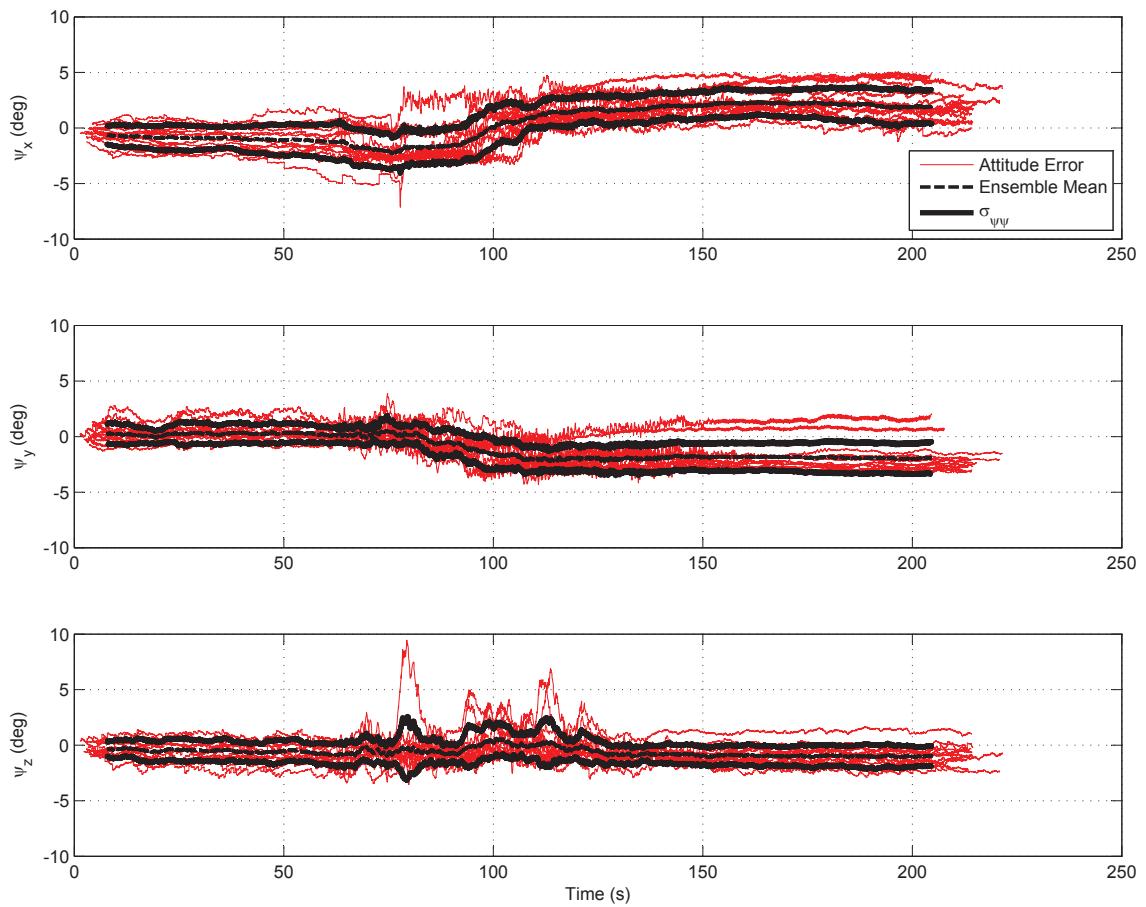
Figure 4.11: Kalman filter ensemble errors. The ensemble of errors in the attitudes estimated by the Kalman filter are shown here with their means and standard deviations.
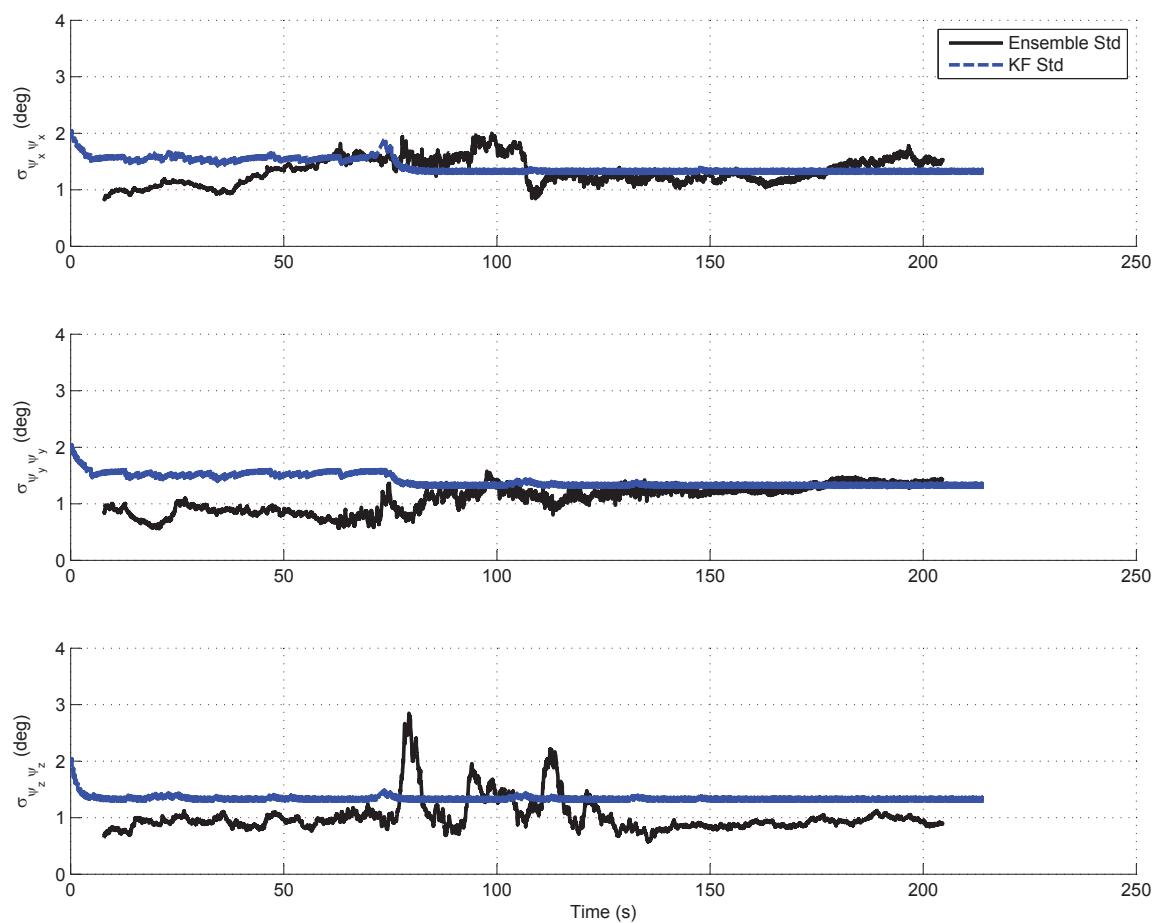
Figure 4.12:   Ensemble and Kalman filer standard deviations. The steady-state standard deviations calculated by the Kalman filter and ensemble standard deviations are close to one another over the entire time segment.

A less stringent requirement and slightly more accurate estimates in the yaw axis indicate that the yaw solution may be sufficiently accurate for such an application.

4.3.2 Measurement Bias. As mentioned before, there is a bias present in the Kalman filter's attitude estimates. This is evidenced by the non-zero ensemble means that can be clearly seen in Figure 4.11. Each of the two sensors used to provide data to the Kalman filter is a possible source of the corruption in the attitude estimates. However, the MEMS inertial sensor's influence can be removed by running the Kalman filter using the tactical inertial measurements without changing the process noise strength matrix, Q. Essentially, this amounts to supplying the filter with as close to a perfect set of inertial measurements as is available, while still allowing the vanishing point measurements to influence the attitude estimates in the same way they had with the MEMS inertial data.

Figure 4.13 shows the ensemble of errors that are obtained when the tactical inertial measurements are used in the Kalman filter. The solutions obtained when using the tactical inertial data in the filter exhibit the same bias that is evident in the filter's solutions using the MEMS inertial measurements. This eliminates the MEMS inertial sensor as the source of the bias, leaving only the optical measurements to blame.

With the the vanishing point measurements pinpointed as the source of the biases, the next step in isolating the cause is to try to find measurements that consistently err in the same direction and with the same magnitude. This was accomplished by visually observing the vanishing point measurements projected onto the images to determine whether erroneous measurements could be identified. The vanishing points affecting the roll estimates cannot be projected onto the image for most of the duration of a test run because they are usually oriented nearly parallel to the image plane, so they were not considered under this review. The largest, sustained bias observed in either the pitch or yaw axes for any one test run is 3 degrees exhibited downward direction of the pitch axis. If vanishing point identified by the detection algorithm
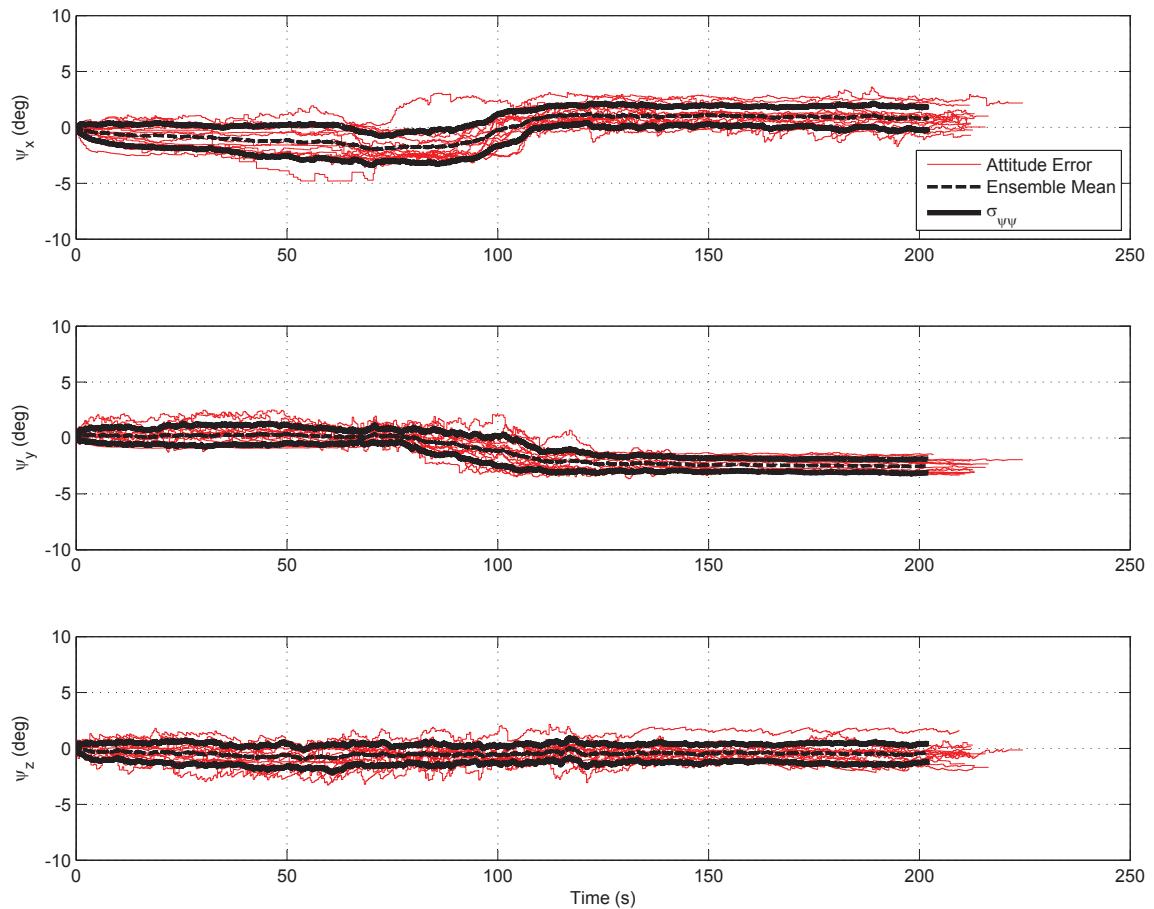
Figure 4.13: Kalman filter ensemble errors using tactical inertial measurements. The ensemble statistics exhibit the same bias that is present in the solutions obtained from the combined MEMS inertial and optical measurements.

were 3 degrees below where the edges of the hallways appear to intersect, it would be noticeable to a human observer examining the image. The processed images with the inlier lines highlighted as in Figures 4.6 and 4.7 from the test run with the 3 degree pitch bias were reviewed and no such measurements were observed.

Time limitations and resource availability did not facilitate any further investigation into the source of the measurement biases. However, other possible causes may include non-orthogonality of the walls, ceilings and floors and straight but non-parallel elements in the environment. Such phenomena violate the assumption that the environment consists primarily of mutually-orthogonal planes and could introduce error into the vanishing point detection and/or attitude estimation process.

4.4   Summary

Shortcomings notwithstanding, the coupling of vanishing point tracking with inertial measurements has been shown to vastly improve vehicle attitude estimates. By joining two inexpensive, lightweight, low-power sensors, a drift-free attitude determination method that is accurate to within a few degrees is attainable.

V. Conclusions and Future Work

This chapter summarizes the information presented earlier in this thesis and provides a few suggestions for how this work could be continued and improved.

5.1 Conclusions

Inertial sensors have been used for decades to provide position and attitude information to various users. Unfortunately, even the most advanced inertial sensors are subject to boundless error growth. The primary objective of this research has been to demonstrate a method for constraining the errors in the attitude solution from a commercial MEMS inertial sensor through the use of computer vision. The two sensors are used in harmony, with the inertial data aiding the vision process and the vision data aiding the inertial process.

The combined visual/inertial attitude estimation method was developed by designing an extended Kalman filter for this purpose. The filter's dynamics model was established using inertial navigation theory. The filter's measurement model was established beginning with the Manhattan world assumption from [7]. A Manhattan world scene contains three primary groupings of parallel lines aligned to the Manhattan grid. The projections of these parallel lines onto the image plane intersect at one of three vanishing points. The directions of the vanishing points reveal the camera's orientation with respect to the scene, and are used to update the Kalman filter's attitude estimates.

A unique method for detecting vanishing points was established which yields unit-length Cartesian 3-vectors indicating each vanishing direction expressed in the camera reference frame. This process utilizes the RANSAC concept from [9] to avoid having to find every possible intersection of image lines. The method is effective at finding vanishing points in many conditions that adhere to the Manhattan world assumption, but some line geometries and measurement noise can give rise to erroneous measurements. Monitoring the measurement residuals and ignoring gross outliers prevents such spurious measurements from influencing the attitude estimates.

89

An experiment was developed to evaluate the extended Kalman filter's attitude estimates. For this experiment, an indoor motion profile mimicking flight was selected along which a MEMS-grade inertial sensor, tactical-grade inertial sensor, and a commercial webcam were transported. This profile guides the sensors through one hallway towards in intersection, clockwise around the corner, and down to the end of a second hallway. Data collected from the MEMS-grade inertial sensor and webcam during fifteen test runs following this profile were post-processed through the extended Kalman filter, and the filter's solutions were compared with the solutions from the tactical inertial sensor to determine their accuracy.

The attitude estimates from the Kalman filter show dramatic improvement over the attitude estimates from the MEMS inertial sensor alone. After only 3.5 minutes of operation, the unaided MEMS inertial estimates have a standard deviation of 26.4 degrees, while the Kalman filter's estimates have standard deviations of 1.5 degrees in the body reference frame's x and y-axes and only 0.9 degrees in the z-axis. Furthermore, the drift rate of 0.63 $\deg^2$/sec in the unaided MEMS estimates is absent from the Kalman filter's solutions.

There is a consistent bias in the Kalman filter's attitude estimates from each of the fifteen test runs. The MEMS inertial sensor was eliminated as a possible cause, leading to the conclusion that the bias originates from the vanishing point measurements. This bias has the greatest magnitude (~2.5 degrees) in the vehicle body reference frame's y-axis.

5.2   Future Work

There are various ways in which the work that has been presented could be extended or improved. Some of these are related to improving or modifying the vanishing point detection process, and others involve extending the attitude estimation research and preparing for on-vehicle implementation.

5.2.1   Vanishing Point Orthogonality.   The attitude estimation technique described in this thesis relies on the assumption that the environment in which a host vehicle is operating consists primarily of many planes and lines oriented in one of three mutually orthogonal directions. The approach to vanishing point detection presented herein consists of looking for the vanishing point in each of the three principal directions individually. However, it may prove effective to search for the complete triad of vanishing points all at once instead of each one in sequence. In [26], Rother presents a computationally intensive approach to finding all three vanishing directions simultaneously in which every possible intersection of two lines from the image is examined. Combining Rother's approach with the RANSAC method presented herein may prove effective at rapidly obtaining all three vanishing directions in an image.

5.2.2   Vanishing Point Detection Robustness.   During one of the test runs that were performed, a pedestrian entered the camera's field of view, walking from behind the camera to the end of the hall and turning the corner. This person's presence in 34 consecutive images appeared to have no impact on the vanishing point measurements obtained from them, even though, when closest to the camera, he or she obstructed approximately 10% of the camera's view. This one example is not sufficient to demonstrate the vanishing point detection method's robustness to such disturbances, nor was making such a determination an objective of this research. However, an investigation into the impact of different amounts and types of obstructions on the vanishing point detection process could be insightful.

5.2.3   Motion Profiles.   This research explored only a single motion profile through a pair of hallways which included a single, 90-degree clockwise change in heading. Additional profiles were not examined due to time and resource limitations. However, investigating other profiles including counter-clockwise turns and entering/exiting rooms adjacent to the hallway could further demonstrate the utility of this attitude estimation method. Furthermore, additional profiles may provide more insight into the cause and nature of the measurement bias discussed in Section 4.3.2.

5.2.4 Real-time Implementation.   The attitude estimates generated during this research effort were all obtained by post-processing the optical and inertial data, but in order to be implemented aboard an aerial vehicle, this task must be accomplished in real time. The microcomputer onboard the intended quadrotor host vehicle has limited computational capability and runs a different operating system than the computer used to process the data. Real-time implementation of the attitude estimation method presented in this thesis would likely require that the Matlab code used to implement the Kalman filter be optimized for faster processing and ported to the C computing language for on-vehicle use. Once these tasks were accomplished, an investigation into the latency in processing images in real time could help determine the feasibility of onboard attitude estimation using the methods presented in this thesis.

5.2.5 Non-Manhattan World.   The attitude estimation method presented in this thesis has been founded on the Manhattan world assumption. While some man-made environments conform to this model, most scenes do not contain many parallel planar surfaces. While images of these non-Manhattan scenes will not contain groups of parallel lines, the concept of a vanishing point is still valid. Identifying the vanishing directions for a non-Manhattan world scene is a much more difficult task, since the intersections of straight lines cannot be used. However, if the vanishing directions were identified, this information could still be combined with inertial measurements to obtain accurate attitude estimates.

5.3 Closing

This research has presented one way in which inertial and optical sensors can be combined to provide improved navigation information. As this technology continues to be developed, an equally-precise alternative to satellite-based navigation may ultimately be achieved. Only time will tell where this field of science will lead.

Bibliography

1. Barnard, Stephen T. "Interpreting perspective images". *Artificial Intelligence*, 21(4):435 – 462, 1983. ISSN 0004-3702.

2. Borkowski, Jeffrey M. and Michael J. Veth. "Passive indoor image-aided inertial attitude estimation using a predictive hough transformation". *Position Location and Navigation Symposium (PLANS), 2010 IEEE/ION*, 295 –302. May 2010. ISSN 2153-358X.

3. Bouguet, Jean-Yves. "Camera Calibration Toolbox for Matlab". URL `http://www.vision.caltech.edu/bouguetj/calib_doc/`.

4. Brown, Duane C. "Close-Range Camera Calibration". *Photgrammetric Engineering*, 37(8):855–866, 1971.

5. Burns, J. Brian, Allen R. Hanson, and Edward M. Riseman. "Extracting Straight Lines". *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, PAMI-8(4):425 –455, July 1986. ISSN 0162-8828.

6. Canny, John. "A Computational Approach to Edge Detection". *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, PAMI-8(6):679 –698, nov. 1986. ISSN 0162-8828.

7. Coughlan, J.M. and A.L. Yuille. "Manhattan World: compass direction from a single image by Bayesian inference". volume 2, 941 –947 vol.2. 1999.

8. Duda, Richard O. and Peter E. Hart. "Use of the Hough transformation to detect lines and curves in pictures". *Commun. ACM*, 15(1):11–15, 1972. ISSN 0001-0782.

9. Fischler, Martin A. and Robert C. Bolles. "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography". *Commun. ACM*, 24(6):381–395, 1981. ISSN 0001-0782.

10. Gonzalez, Rafael C. and Richard E. Woods. *Digital Image Processing*. Pearson Prentice Hall, 3rd edition, 2008.

11. Hartley, Richard and Andrew Zisserman. *Multiple View Geometry*. Cambridge University Press, second edition edition, 2003.

12. Hough, Paul V. C. "Method and Means for Recognizing Complex Patterns", U.S. Patent 3069654, December 1962.

13. Johnson, Neil. *Vision-Assisted Control of a Hovering Air Vehicle in an Indoor Setting*. Master's thesis, Brigham Young University, August 2008.

14. Kahn, Philip, Leslie Kitchen, and Edward Riseman. *Real-Time Feature Extraction: A Fast Line Finder for Vision-Guided Robot Navigation*. Technical Report 87-57, University of Massachusetts, Amherst, July 1987.

15. Kalman, Rudolf E. "A New Approach to Linear Filtering and Prediction Problems". *Journal of Basic Engineering*, 82:35–45, 1960.

16. Kemp, Christopher. *Visual Control of a Miniature Quad-Rotor Helicopter*. Ph.D. thesis, Churchill College, University of Cambridge, February 2006.

17. Kessler, Christoph, Christian Ascher, Natalie Frietsch, Michael Weinmann, and Gert F. Trommer. "Vision-based attitude estimation for indoor navigation using Vanishing Points and lines". *Position Location and Navigation Symposium (PLANS), 2010 IEEE/ION*, 310 –318. May 2010. ISSN 2153-358X.

18. Kiryati, N., Y. Eldar, and A.M. Bruckstein. "A probabilistic Hough transform". *Pattern Recognition*, 24(4):303 – 316, 1991. ISSN 0031-3203.

19. Košecká, Jana and Wei Zhang. "Video Compass". *ECCV '02: Proceedings of the 7th European Conference on Computer Vision*, 476–490. 2002.

20. Magee, M.J. and J.K. Aggarwal. "Determining Vanishing Points from Perspective Images". *Computer Vision, Graphics, and Image Processing*, 26:256–267, 1984.

21. Maybeck, Peter S. *Stochastic Models, Estimation, and Control*, volume I. Navtech Book & Software Store, 1994.

22. Maybeck, Peter S. *Stochastic Models, Estimation, and Control*, volume II. Navtech Book & Software Store, 1994.

23. Norton Schwartz, USAF, General. "Keynote Address". *38th IFPA-Fletcher Conference on National Security Strategy and Policy*. January 2010.

24. Prewitt, J. M. S. "Object Enhancement and Extraction". *Picture Processing and Psychopictorics*, 1970.

25. Roberts, L. G. *Machine Perception of Three-Dimensional Solids*. Technical Report TR315, Massachusetts Institute of Technology, Lexington Lincoln Laboratory, May 1965.

26. Rother, Carsten. "A new approach to vanishing point detection in architectural environments". *Image and Vision Computing*, 20(9-10):647 – 655, 2002. ISSN 0262-8856.

27. Schuster, R., N. Ansari, and A. Bani-Hashemi. "Steering a robot with vanishing points". *Robotics and Automation, IEEE Transactions on*, 9(4):491 –498, aug. 1993. ISSN 1042-296X.

28. Sobel, Irwin E. *Camera Models and Machine Perception*. Ph.D. thesis, Stanford University, 1970.

29. Titterton, David and John Weston. *Strapdown Inertial Technology*. The Institution of Engineering and Technology, second edition, January 2005.

30. Van Loan, Charles F. "Computing integrals involving the matrix exponential". *Automatic Control, IEEE Transactions on*, 23(3):395 – 404, June 1978. ISSN 0018-9286.

31. Veth, Michael J. *Fusion of Imaging and Inertial Sensors for Navigation.* Ph.D. thesis, Air Force Institute of Technology, 2006.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704–0188

1. REPORT DATE *(DD–MM–YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From — To)*
24–03–2011	Master's Thesis	Sept 2009 — Mar 2011

4. TITLE AND SUBTITLE

Coupling Vanishing Point Tracking
with Inertial Navigation to
Estimate Attitude in a Structured Environment

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Dayvid Prahl, Capt, USAF

5d. PROJECT NUMBER

None

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Air Force Institute of Technology
Graduate School of Engineering and Management (AFIT/EN)
2950 Hobson Way
WPAFB OH 45433-7765

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT/GE/ENG/11-33

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

INTENTIONALLY LEFT BLANK

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT

13. SUPPLEMENTARY NOTES

14. ABSTRACT

This research aims to obtain accurate and stable estimates of a vehicle's attitude by coupling consumer-grade inertial and optical sensors. This goal is pursued by first modeling both inertial and optical sensors and then developing a technique for identifying vanishing points in perspective images of a structured environment. The inertial and optical processes are then coupled to enable each one to aid the other. The vanishing point measurements are combined with the inertial data in an extended Kalman filter to produce overall attitude estimates. This technique is experimentally demonstrated in an indoor corridor setting using a motion profile designed to simulate flight. Through comparison with a tactical-grade inertial sensor, the combined consumer-grade inertial and optical data are shown to produce a stable attitude solution accurate to within 1.5 degrees. A measurement bias is manifested which degrades the accuracy by up to another 2.5 degrees.

15. SUBJECT TERMS

vision-aiding, inertial navigation, vanishing points, attitude

16. SECURITY CLASSIFICATION OF:

a. REPORT	b. ABSTRACT	c. THIS PAGE
U	U	U

17. LIMITATION OF ABSTRACT

UU

18. NUMBER OF PAGES

109

19a. NAME OF RESPONSIBLE PERSON
Michael J. Veth, LtCol, USAF

19b. TELEPHONE NUMBER *(include area code)*
(850) 882-4667; michael.veth@eglin.af.mil

Standard Form 298 (Rev. 8–98)
Prescribed by ANSI Std. Z39.18